



MODELIZACIÓN DE UN PROTOCOLO DE NIVEL II CON REPETICIÓN SELECTIVA OPTIMIZADA USANDO REDES DE PETRI

Proyecto de Software

Director

Ing. LUIS A. MARRONE

Autoras

TES
97/18
DIF-01990
SALA



UNIVERSIDAD NACIONAL D
FACULTAD DE INFORMATICA
Biblioteca
50 y 120 La Plata
catalogo.info.unip.edu.ar
biblioteca@info.unip.edu.ar



DIF-01990

MARISA SPOSATO

MARÍA ELENA OKOSHI



Prefacio

La amplia revolución ocurrida en el campo de las comunicaciones de datos afectó enormemente a las redes de computadoras que evolucionaron a partir de su concepción como una curiosidad académica hasta llegar a ser en la actualidad un elemento más dentro de los servicios informáticos. Se ve reflejado en la popularidad que adquirieron las LAN, MAN y WAN. Un ejemplo de ello es la red Internet. Además los sistemas operativos se perfeccionaron de tal modo que soportan las interfases que permiten la utilización de redes.

Por lo descripto anteriormente, el estudio del tema nos ha llevado a presentar un proyecto que se centra en la definición de un protocolo que optimice la utilización del medio físico (ancho de banda) mediante la repetición selectiva de tramas. Su especificación y análisis de los alcances se efectúa a través de redes de Petri.

Contenido de los Capítulos

En el Capítulo I se efectúa una introducción a la Capa de Enlace haciendo hincapié en los protocolos de ese nivel.

El Capítulo II está íntegramente dedicado a las propuestas definidas, es decir al funcionamiento de los protocolos y estructura de tramas.

El Capítulo III se refiere a la especificación que se realiza con Redes de Petri de los aspectos más salientes de la propuesta seleccionada.

La segunda parte del trabajo está dividida en tres apéndices. En el A se incorporan ejemplos de las corridas con sus respectivas explicaciones.

En el Apéndice B se muestran los dos algoritmos de las propuestas definidas anteriormente.

Como Apéndice C se incluye el programa TESIS.SIM correspondiente a la Red de Petri, el cual fue editado bajo SIMNET v.1.35. El mismo tiene una presentación vectorial y otra gráfica la cual se adjunta al final.

Además se acompaña un diskette conteniendo TESIS.SIM y el simulador SIMNET v.1.35.

Agradecimientos

Deseamos agradecer a Luis Marrone por su dedicación y su innegable vocación docente.

A mis padres por todo el apoyo que me brindaron.

A mi compañera de tareas.

Marisa Sposato

A Marisa Sposato por ser “ la otra mitad del trabajo “ y por su tolerancia.

A mis padres por toda la energía que me transmiten ...

María Elena Okoshi

La Plata

Diciembre 1996.

Contenido

Prefacio	I
----------------	---

Capítulo I : Visión General de la Capa de Enlace

1. Importancia de la Capa de Enlace en el Modelo OSI	1
1.1 Protocolos de la Capa de Enlace	2
1.2 Breve Reseña de un Protocolo Típico de Nivel 2: HDLC	8
1.3 Medios de Transmisión y Protocolos	9

Capítulo II : Desarrollo de Propuestas

2. Conceptos Básicos	11
2.1 Enlace de Datos	11
2.2 Configuración	12
2.3 Gestión de Enlace	12
2.3.1 Establecimiento del Enlace	13
2.3.2 Transferencia de información	15
2.3.2.1 Propuesta I	16
2.3.2.1.1 Tabla de tramas	16
2.3.2.1.2 Desarrollo	17
2.3.2.2 Propuesta II	27
2.3.2.2.1 Tabla de tramas	27
2.3.2.2.2 Desarrollo	28
2.3.3 Desconexión del Enlace	36
2.4 Estructura de la Trama	36
2.4.1 Propuesta I	37
2.4.2 Propuesta II	42
2.5 Análisis Comparativo de las Propuestas	44
2.6 Conclusión	54

Capítulo III : Especificación con Redes de Petri

3. Especificación del Protocolo	56
3.1 Generalidades	56
3.2 Modelización del Protocolo	57
3.2.1 Envío de Paquetes	57
3.2.2 Envío de Tramas	58
3.2.3 Timer de Emisión	59
3.2.4 Llegada de Tramas y Timer de Recepción	60
3.2.5 Confirmación	61
3.2.6 Puesta en Línea de la Confirmación	62
3.2.7 Llegada de la Confirmación al Emisor	63
3.2.8 Indicadores de Confirmación	64
3.2.9 Pedido de Retransmisión	65
3.2.10 Pedido de Retransmisión en Línea y Llegada	67
3.2.11 Envío del REQ	68
3.2.12 Respuesta ante un REQ	69
3.2.12.1 Pedido de Retransmisión	70

3.2.12.2 Confirmación	71
3.2.12.3 Retransmisión de Tramas	73
3.2.13 Temporizaciones	74
3.2.14 Piggybacking	74
3.3 Simulación	75

Capítulo IV : Conclusiones Finales

4. Conclusiones Finales	77
-------------------------------	----

Capítulo V : Bibliografía

5. Bibliografía	79
-----------------------	----

Capítulo VI : Glosario

6. Glosario	81
-------------------	----

Apéndice A

A. Resultados	84
A.1 Ejemplo 1: Transmisión Normal	84
A.2 Ejemplo 2.1: Pérdida de una trama de información (Ventana 2)	86
A.3 Ejemplo 2.2: Pérdida de dos Tramas de Información (Ventana 3)	90
A.4 Ejemplo 3: Pérdida de Tramas de Información de la Ventana Completa	92
A.5 Ejemplo 4: Pérdida de la Confirmación	96
A.6 Ejemplo 5: Miscelánea I	100
A.7 Ejemplo 6: Miscelánea II	104
A.8 Ejemplo 7: Solapamiento de Tiempos	110
A.9 Ejemplo 8: Mala Configuración de Timers	114

Apéndice B

B. Algoritmos	119
B.1 Propuesta I	119
B.2 Propuesta II	130

Apéndice C

C. Tesis.Sim	142
--------------------	-----

Capítulo I

VISIÓN GENERAL DE LA CAPA DE ENLACE

1. Importancia de la Capa de Enlace en el Modelo OSI

Tomando como referencia el modelo OSI, el nivel 2 lo constituye la Capa de Enlace. El servicio de la misma es fundamentalmente, el de proveer una línea de transmisión que se presente libre de errores, independientemente de la línea física existente entre dos nodos contiguos. La Capa de Red que usa este servicio dispone de un canal virtual capaz de transmitir sus unidades de información de forma confiable y eficiente. Además en toda transmisión existe una trayectoria real de datos como se muestra en la fig. 1.

Dado que la Capa Física acepta y transmite un flujo de bits sin tener en cuenta su semántica y estructura, la Capa de Enlace debe crear y reconocer los límites de cada trama

La línea física está sujeta a diversos tipos de interferencia y mal funcionamiento que resultan en errores de transmisión. Corresponde a la Capa de Enlace resolver los problemas causados por daño, pérdida o duplicidad de tramas.

Cuando los tiempos del emisor y receptor no están sincronizados se pueden dar una serie de situaciones distintas. La que resulta perjudicial es aquella en la que el emisor envía tramas a una tasa mayor que la que puede procesar el receptor.

Los datos dentro de un canal de comunicaciones tienen una velocidad de transmisión finita y un retardo de propagación que varía según el medio. Esto, unido a la velocidad finita que tiene el nodo en procesar los datos determina la eficiencia en la transmisión de los mismos. Si la línea es full-duplex el flujo de datos del emisor compite con los del receptor.

A raíz de estos problemas surge la necesidad de formular un protocolo que permita la resolución de los mismos a nivel de la Capa de Enlace.

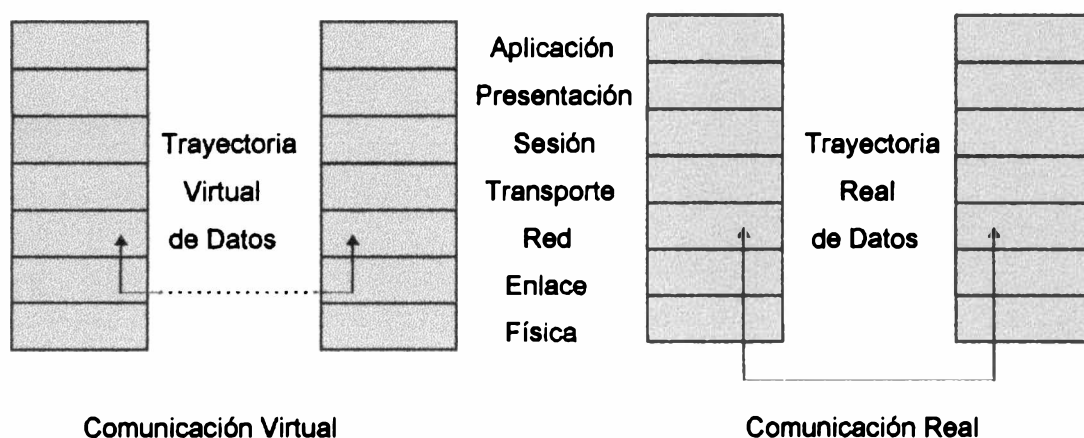


Fig.1 - Comunicación



1.1 Protocolos de la Capa de Enlace

El modelo de comunicación se basa en ciertas premisas.

Los procesos de la Capa de Red, la de Enlace y la Física son independientes y se intercomunican mediante el envío de mensajes.

El emisor transmite sus datos al receptor usando un servicio de conexión confiable.

La Capa de Enlace del emisor toma el paquete de la Capa de Red y le agrega un header y un trailer como dato de control. La trama ya constituida pasa a la Capa Física para ser transmitida al receptor. El conjunto de bits recibido por la Capa Física del receptor pasa a la Capa de Enlace donde se establecen los límites de la trama y el hardware dedicado, analiza el código de redundancia. Si la trama llega sin daño verifica si su número está en orden, entonces pasa el paquete a la Capa de Red.

El conjunto de reglas y convenciones entre entidades de la capa se conoce como protocolo de la Capa de Enlace. Es una abstracción lógica del proceso físico de comunicación.

Se han desarrollado un número de protocolos que parten de ciertos supuestos. Uno de ellos considera una línea libre de errores. Si el medio es confiable, es decir el canal no pierde ni daña tramas, éstas estarán constituidas por un campo que las delimita y el campo de datos. La transmisión se hace unidireccionalmente y está orientado a un servicio sin conexión. En este caso el emisor transmite datos indefinidamente y la Capa de Red del receptor los procesa a una velocidad infinitamente grande.

Pero, en realidad el espacio de memoria para el almacenamiento de tramas de la Capa de Enlace del receptor es limitado. Con lo que se suscita el problema de que el emisor sature al receptor al transmitir datos a una velocidad superior de la que el destinatario sea capaz de procesar. Una solución sería programar de modo de tener un emisor que se ajuste a los tiempos del receptor. Aunque generalmente, la Capa de Enlace del receptor no tiene una dedicación exclusiva para una sola línea sino que se ocupa de varias por lo que los tiempos de llegada y procesamiento pueden ser muy variables. Un planteo hipotético sería considerar el peor caso calculando el retardo máximo para evitar la inundación por parte del transmisor. Esto llevaría a una baja utilización del ancho de banda. Una respuesta más adecuada a esta situación consistiría en realizar una retroalimentación del receptor con el fin de confirmar la trama enviada por el emisor. El mismo no podrá transmitir una nueva trama mientras espera la llegada de la confirmación. Este protocolo se conoce como Stop and Wait. Bajo estas circunstancias se requiere un canal half-duplex con transmisión bilateral alternada. El asentimiento no necesitará ser analizado por el emisor, a consecuencia de estar frente a un canal libre de errores.

Sin embargo, por lo general los protocolos hacen este control en la Capa de Enlace. La solución estaría en colocar en el emisor un temporizador. Al enviarse una trama y no recibir su

correspondiente asentimiento por llegar dañada o no haber sido reconocida por el receptor, una vez expirado el temporizador se debe retransmitir la misma.

Si bien este esquema resulta ser necesario no es suficiente, ya que si existe ruido en el canal puede afectar independientemente a las tramas de datos como a las de control. Es decir, si se envía una trama y llega correctamente al receptor, éste genera un asentimiento que puede no ser recibido por el emisor. En tal caso el transmisor, vencido el temporizador vuelve a poner en línea la misma trama. Como el receptor no tiene forma de comprobar que las dos tramas recibidas sucesivamente son iguales, pasa ambas a la Capa de Red. Para salvar la duplicidad de tramas basta con colocar un número de secuencia de un bit (0 o 1) como parte de la cabecera a modo de identificación de la trama.

Si se envía una trama que llega correctamente al receptor y el asentimiento no llega al emisor, éste volverá a retransmitir la misma trama. El receptor intuirá que esa trama fue enviada anteriormente por su número de secuencia, por lo tanto desechará la misma, mandando nuevamente el asentimiento.

Estamos en presencia de un protocolo que también transmite datos en una sola dirección. Si bien soporta la recuperación de tramas perdidas, se necesita que el plazo de vencimiento del temporizador sea lo suficientemente largo para evitar que el emisor retransmita una trama mientras el asentimiento está aún viajando.

Cuando por fin llega el asentimiento al emisor, éste cree que se le confirmó la trama últimamente enviada, por lo tanto transmite una nueva cambiando el número de secuencia. Si ésta se perdiese completamente, pero el asentimiento extra arriba de manera correcta, el emisor no retransmite la trama extraviada, en cuyo caso este protocolo falla.

En los protocolos descritos se transmite en una sola dirección. En la práctica se necesita transmitir en ambas direcciones.

Una solución sería tener dos canales físicos separados. El problema que acarrea es el desperdicio casi total del ancho de banda del canal contestador.

Otra, sería utilizar el mismo circuito intercalando tramas de datos y de asentimientos.

Sin embargo, se podría obtener una mejora retardando la emisión del asentimiento con el fin de conseguir transporte gratuito con la próxima salida de la trama de datos del receptor al emisor. Esta técnica se conoce como piggybacking.

La ventaja principal que brinda el piggybacking es una mejor utilización del ancho de banda disponible en el canal.

La desventaja está dada por los tiempos de espera de la Capa de Enlace del receptor en cuanto a los paquetes procedentes de la Capa de Red y su sincronía con el temporizador del emisor.

La necesidad en la búsqueda de protocolos que soportaran sincronización a pesar de la existencia de tramas perdidas o dañadas y temporizaciones expiradas prematuramente, dio origen a una clase denominada de ventana deslizante.

En estos protocolos, cada trama del emisor contiene un número de secuencia que pertenece al rango 0 hasta un valor máximo. Este valor se calcula como $2^n - 1$, con lo cual el número entra en un campo de n bits.

El emisor mantiene una lista de números de secuencia consecutivos correspondientes a las tramas que desea enviar. Estas están dentro de la ventana emisora. De igual modo el receptor administra las tramas que puede recibir en su ventana receptora.

En el emisor permanecerán todas las tramas de las cuales no ha recibido asentimiento. La Capa de Red envía paquetes a la Capa de Enlace, ésta le asigna el siguiente número de secuencia, lo pone en su ventana y lo envía, mientras no supere el tamaño máximo de la ventana. En tal caso esta capa desactiva a su superiora.

Cualquier trama que cayese fuera de la ventana receptora será descartada.

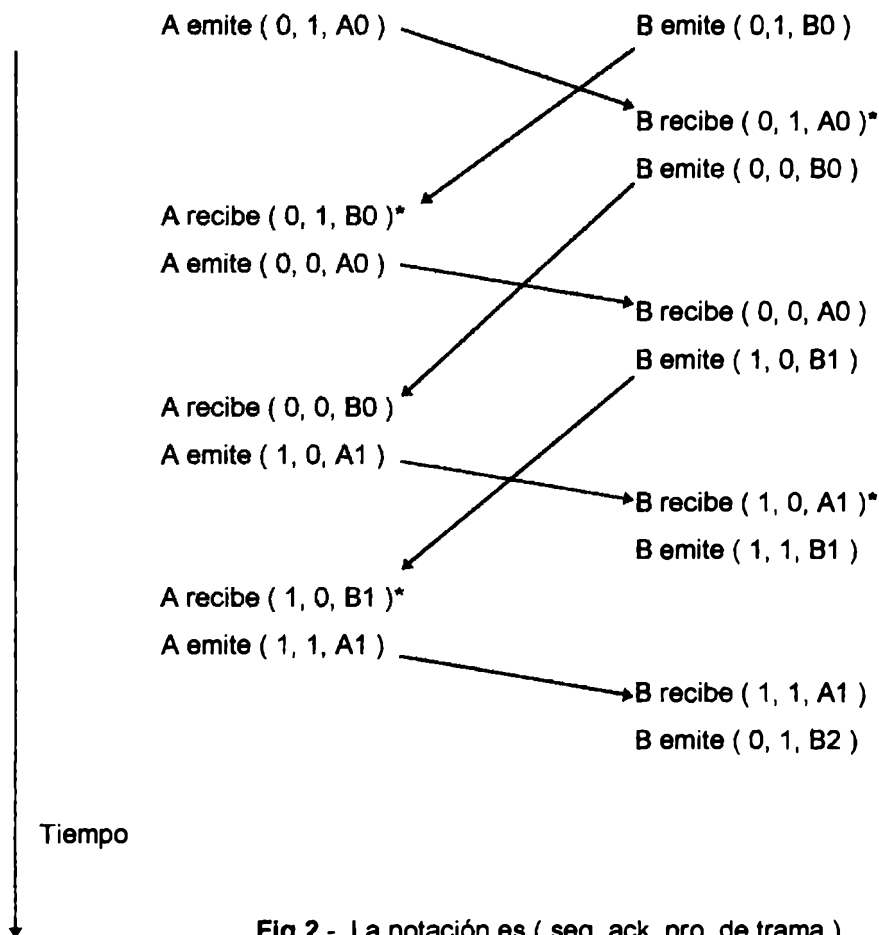


Fig.2 - La notación es (seq, ack, nro. de trama)

El (*) indica el lugar donde la Capa de Red acepta un paquete

Cada trama recibida donde el número de secuencia corresponde al borde inferior de la ventana pasa a la Capa de Red, se genera el asentimiento y la ventana se rota en uno.

El protocolo de ventana deslizante de tamaño máximo de uno utiliza el concepto de Stop and Wait. La Capa de Enlace solo acepta tramas ordenadas. Cuando una trama se pierde por ruido en la línea, no podrá ser reconocida por el receptor. En tal caso no podrá responder ni con un asentimiento positivo (Ack) ni con uno negativo (NAck). Para contemplar esta posibilidad el emisor cuenta con un temporizador, el cual se le vence y retransmite la misma trama.

A pesar de que ninguna combinación de tramas perdidas o vencimientos prematuros cause la duplicación, la pérdida de paquetes o un posible deadlock, surge una situación particular cuando ambos lados envían un paquete inicial simultáneamente, como se ilustra en la figura 2. La mitad de las tramas contienen duplicados, aún cuando no haya errores de transmisión.

Hasta ahora se considera que los tiempos de transmisión de la trama y del asentimiento eran despreciables, pero en la realidad muchas veces esto no es así. Si se toman en cuenta que los tiempos de propagación son altos como en el caso de la transmisión satelital, el emisor estará gran parte del tiempo ocioso esperando el asentimiento antes de enviar una nueva trama.

La solución radica en no enviar una trama por vez, sino un total de w , antes de que se bloquee. Se puede elegir w de tal forma que el emisor envíe tramas continuamente sin que se sature la ventana. Esta técnica se conoce como pipelining.

Si se cuenta con un canal no confiable, el pipelining puede generar algunos problemas. Si se envían varias tramas y una de ellas llega dañada se debe determinar el tratamiento de las siguientes tramas que llegaron bien.

Para ello contamos con dos soluciones. Una, en la llamada Go-Back-N, la estación emisora podrá enviar una cantidad de tramas dependiendo del tamaño de la ventana. Si el receptor detecta error en una trama envía un NAck para la misma. La Capa de Enlace acepta solo la siguiente trama esperada por la Capa de Red. Es decir, el receptor descartará todas las tramas sucesivas hasta que la trama en cuestión sea recibida correctamente. En consecuencia el emisor cuando recibe el Nack deberá retransmitir la trama, además de todas las sucesivas.

Las tramas no necesitan ser confirmadas individualmente, pero la ventana del receptor no podrá ser superior a $(2^n - 1)$ mientras el espacio intersecuencial es de 2^n .

Si bien el piggybacking trae aparejado un beneficio substancial al enviarse un asentimiento conjuntamente con una trama de información, podría traer serias consecuencias en el supuesto caso en que contásemos con una ventana y espacio intersecuencial de 2^n . Según el siguiente ejemplo si se tiene ventana 8 y secuencias de 0 a 7, es decir ocho tramas, podría suceder que el emisor transmitiese la trama 0, el receptor confirmaría con una trama de información con Ack 0, según la fig. 3.1. Posteriormente se emiten 1, 2, 3, ..., 7, 0, a las cuales el receptor confirmaría con otra trama de información con Ack 0, fig. 3.2. Esto significaría que las

ocho tramas llegaron correctamente, pero también podría indicar que las 8 se perdieron en el tránsito y la estación receptora estaría repitiendo un Ack 0, como lo muestra la fig. 3.3.

Como el piggybacking inexorablemente necesita de algún número de trama para la confirmación, Go-Back-N podría ser soportado con una ventana de tamaño $2^n - 1$, teniendo un espacio intersecuencial de 2^n .

Cabe destacar que en los esquemas con piggybacking, al tener la trama un campo de asentimiento hace que si ésta se perdiese con la próxima se podrían confirmar ambas tramas.

Cada trama posee un temporizador propio que expira independientemente de los demás.

Este protocolo es apropiado para canales de comunicación seguros, donde los errores sean esporádicos, ya que en una línea ruidosa la retransmisión de tramas causa un desperdicio considerable del ancho de banda.

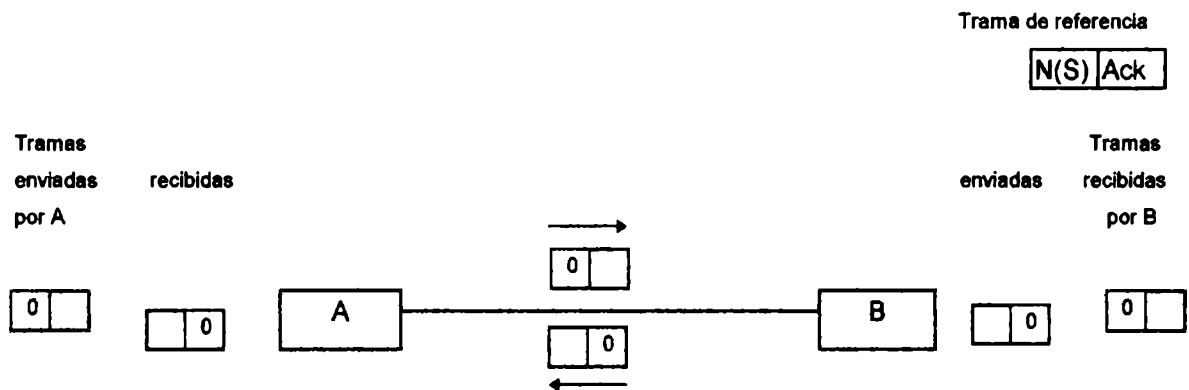


Fig. 3.1 - Confirmación de la trama 0

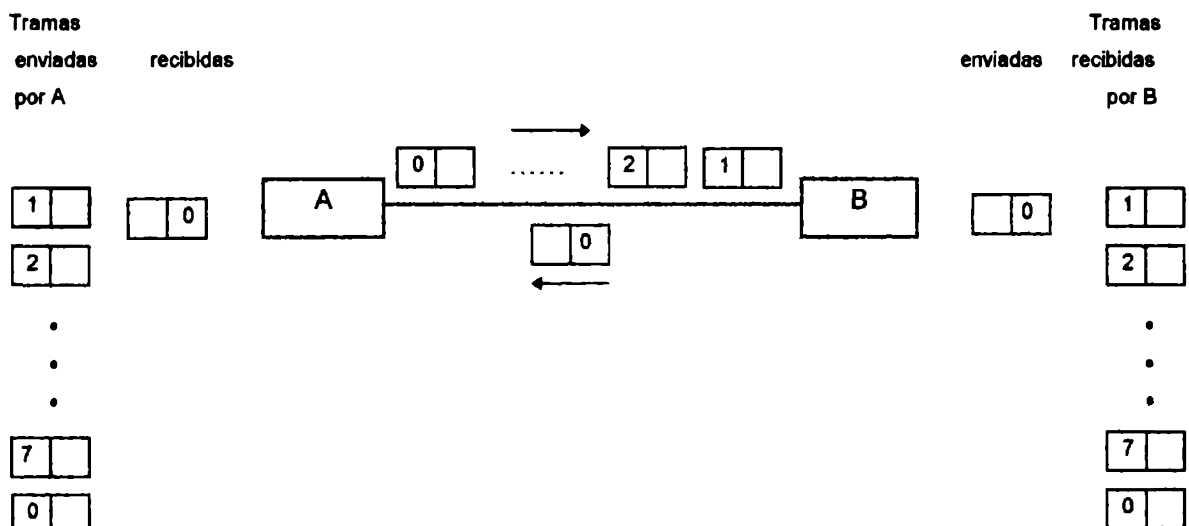


Fig. 3.2 - Arribo de las ocho tramas al receptor y posterior asentimiento

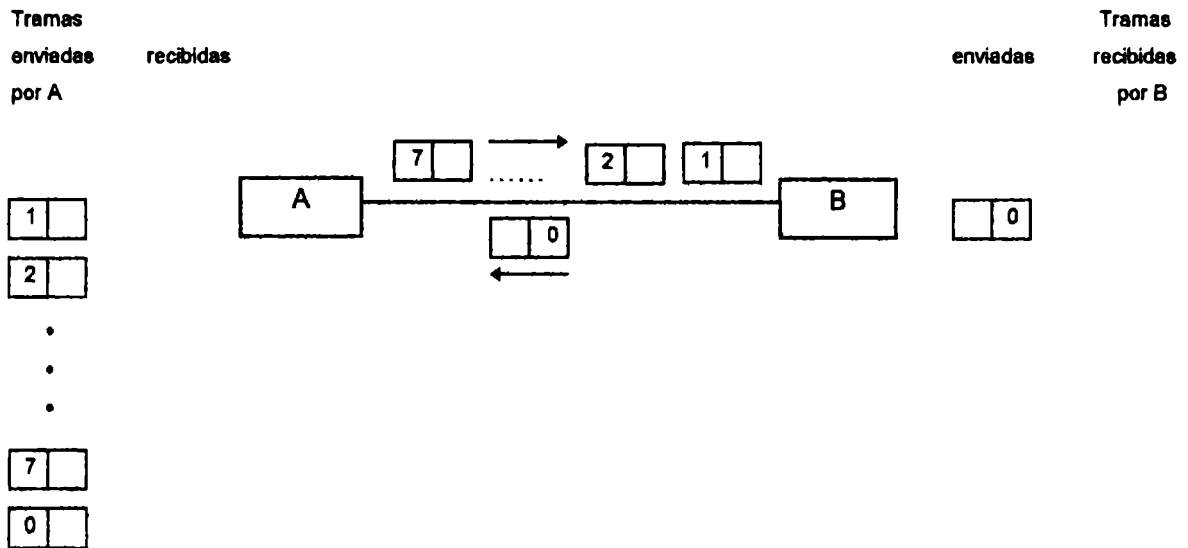


Fig. 3.3 - Retransmisión del Ack perteneciente a la trama 0

La otra solución es conocida como Repetición Selectiva. Provee una aproximación más refinada que Go-Back-N. Solo se retransmiten las tramas que recibieron un NAck y el receptor deberá estar capacitado para almacenar las tramas posteriores a la dañada hasta que sea retransmitida. Además poseer la lógica suficiente que permita reinsertar la trama en la secuencia correspondiente. A su vez, el transmisor deberá estar preparado para enviar tramas fuera de secuencia.

La Repetición Selectiva genera mayores controles que el Go-Back-N, lo que lo hace más complejo pero a la vez es más eficiente al permitir un mejor aprovechamiento del enlace.

En un medio donde los retardos no son despreciables, una aproximación como Go-Back-N desaprovecharía el ancho de banda mientras que la Repetición Selectiva brinda mayor performance.

El hecho de enviar y recibir tramas fuera de secuencia hace que surjan ciertos problemas. Uno de ellos es el que está relacionado con el tamaño de ventana. Si el número de secuencia ocupa 3 bits, el emisor podrá enviar hasta un máximo de $(2^3 - 1)$ tramas sin asentimiento previo. Es decir, de la 0 a la 6 la primera vez. El receptor está en condiciones de aceptar cualquier trama cuyo número de secuencia esté entre dichos números. Si las tramas llegasen correctamente a destino, el receptor las confirmará y avanzará su ventana para permitir la recepción de las tramas 7, 0, 1, 2, 3, 4 y 5. Si el Ack no llegase al emisor, se vencerán los temporizadores paulatinamente con lo cual se retransmitirán las tramas de la 0 en adelante. Debido a que el receptor acepta cualquier trama cuyo número de secuencia cae en su ventana aceptará este grupo de tramas

como nuevas. El protocolo falla porque el receptor no distingue entre tramas transmitidas y retransmitidas.

La solución a esta cuestión consiste en que no se solapen ambas ventanas, del emisor y del receptor. Es decir, considerar el tamaño máximo de la ventana como la mitad del rango de los números de secuencia, de forma tal de determinar si el lote de tramas enviado son duplicadas o nuevas.

La técnica del piggybacking no siempre puede ser buena ya que si el receptor no tiene tramas de datos para enviar el asentimiento se retendrá por un largo tiempo. Este problema se ve resuelto al activar en el receptor un temporizador al llegar una trama de datos en secuencia. Si no hay datos para enviar del receptor al emisor, antes que venza se enviará una trama de asentimiento solamente.

1.2 Breve Reseña de un Protocolo Típico de Nivel 2: HDLC

Los protocolos orientados a bit satisfacen una serie de requerimientos como ser: enlaces punto a punto y multipunto, transmisión half-duplex y full-duplex, modalidad primaria-primaria y primaria-secundaria y, soportan distancias cortas y largas.

HDLC (High-Level Data Link Control), define tipos de estaciones primaria, secundaria y combinada; configuraciones de enlace balanceada y desbalanceada. Opera con los modos de transferencia NRM, ABM y ARM.

Usa la transmisión sincrónica. Todas las comunicaciones se efectúan mediante el intercambio de tramas de datos o de control de formato único.

La estructura de la trama es como la que se muestra en la fig. 4. La longitud mínima es de 32 bits excluyendo los flags.

La trama está delimitada entre dos secuencias de flag cuyo formato es el siguiente: (01111110).

El campo de Dirección se utiliza para identificar a la estación secundaria a la cual transmite o de la cual espera recibir tramas. En enlaces punto a punto este campo no se haría necesario, sin embargo se la utiliza por motivos de uniformidad.

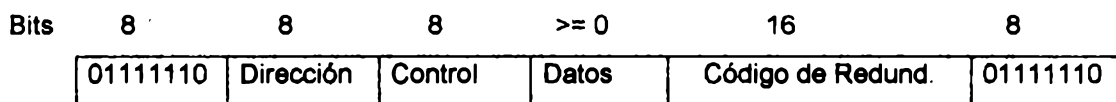


Fig. 4.- Estructura de la trama

Bits	1	3	1	3
Información	0	N (S)	P/F	N (R)
Supervisora	1	0	Tipo	P/F
No numerada	1	1	Tipo	Modificador

Fig. 5 - Campo de Control

El formato del campo de Control indicada en la fig. 5, señala que se trata de una trama de información, supervisora o no numerada

Una trama de información contiene los datos del usuario y su número de secuencia, además de la confirmación positiva de las tramas recibidas dada por el piggybacking.

La trama supervisora realiza el control de flujo y de errores. Una trama puede ser confirmada positivamente con RR. Cuando el receptor no está en condiciones de aceptar más tramas, frena el envío a través de un RNR. Debido a que permite REJ y SREJ, realiza un pedido de retransmisión en base al algoritmo de Go-Back-N o de Repetición Selectiva respectivamente.

Las tramas no numeradas se emplean para una serie de funciones de control. Estas tramas no llevan número de secuencia, por lo tanto no alteran el flujo de tramas de información. Se las utiliza para el establecimiento, reinicialización y desconexión del enlace. Además reporta ante alguna situación irrecuperable en el procesamiento normal de la transferencia de información.

El Código de Redundancia (CRC) es un código matemático que permite chequear errores. El polinomio representado por la trama más el código de redundancia debe ser divisible por un polinomio llamado generador, el cual es conocido por el emisor y el receptor. Una vez recibida la trama, el receptor la divide por el polinomio generador. Si el resto es distinto de cero la trama en cuestión tiene error.

1.3 Medios de Transmisión y Protocolos

Una transmisión de datos depende principalmente de dos factores: la calidad de la señal transmitida y las características del medio. La misma se lleva a cabo a través de un cable como ser coaxial, par trenzado, fibra óptica o puede que la señal no esté contenida físicamente como ocurre con las comunicaciones por aire.

Uno de los aspectos a tener en cuenta en la transmisión de datos es el retardo que puede acarrear un medio. Los enlaces terrestres de microondas tienen un retardo de propagación de

aproximadamente $3\mu\text{s}/\text{km.}$, mientras para los de cable coaxial es de $5\mu\text{s}/\text{km.}$, ya que las señales electromagnéticas viajan a una velocidad menor por un alambre de cobre que por el aire.

A menudo, se dice que los enlaces satelitales sufren un mayor retardo que los terrestres; si bien el retardo de propagación es mayor, el retardo total depende también del ancho de banda y de la tasa de error. Por ej. el retardo total para enviar X Kbits de información sobre una línea terrestre a una velocidad de 9600 bps es de $X/9.6$ seg. Si quisiésemos enviar el mismo mensaje por vía satélite a una velocidad de 5 Mbps, se necesitarían $[(X/5000) + 0.270]$ seg. incluyendo el retardo de propagación que típicamente es de 270 ms. Para los mensajes cuya longitud supera los 2.6 Kbits, resulta más rápido el envío por satélite.

Usando el protocolo de Stop and Wait sobre un enlace satelital, la eficiencia del mismo prácticamente es nula. Un ejemplo de ello está dado por el siguiente caso: un servicio de transmisión de 56 Kbps y tramas de 4000 bits, lo que implica un tiempo de transmisión de $4000/56000 = 71$ ms. La razón entre el tiempo de propagación y el de transmisión es de $270/71 = 3.8$. La utilización máxima equivale a $1/(1 + 2 \times 3.8) = 0.12$. En consecuencia la utilización del protocolo Stop and Wait conduce a un bajo rendimiento considerando un medio como el satelital donde los retardos son notables frente a los tiempos de transmisión actuales.

En general el protocolo de Stop and Wait conduce a una mala utilización de la línea haciéndose más evidente en los casos donde el retardo es importante.

Una mejor utilización del canal puede conseguirse a través del protocolo de Ventana Deslizante en donde se pueden confirmar varias tramas por vez.

Otro elemento a tener en cuenta en la transmisión de datos es el ruido. Una característica es que, generalmente se da en ráfagas afectando un conjunto de tramas.

Existen protocolos como HDLC que tienen mecanismos que posibilitan el pedido de retransmisión de tramas. Se realiza de a una por vez teniendo que enviar el receptor tantos pedidos como tramas solicite.

Teniendo en cuenta la influencia del medio en una transmisión y considerando el alto grado del retardo de propagación que pueden tener algunos medios, los protocolos existentes hacen una pobre utilización del ancho de banda. A fin de lograr optimizarlos se pensó en un protocolo que solucione estas falencias.

Capítulo II

DESARROLLO DE PROPUESTAS

2. Conceptos Básicos

Se utiliza la disciplina orientada al bit conduciendo a un protocolo que cuente con características relevantes como la alta confiabilidad y eficiencia dada por los procedimientos de detección y recuperación de errores, así como el control de error y flujo mediante un overhead mínimo en la trama. Además de ello la independencia de código permite que se utilice un conjunto de bits para transmitirse.

La clave para satisfacer los requerimientos consiste en tener significancia posicional y campos de control codificados. Una estructura es posicionalmente significativa cuando está dividida en campos, cada uno de los cuales tiene un significado y cuya posición está relacionada con un delimitador de la trama. Un campo de control codificado es un campo en el cual las diferentes combinaciones de bits tienen significados específicos y donde los subcampos tienen sentido.

En cuanto al enlace, la configuración utilizada es punto a punto, y la transmisión de información puede ser enviada en ambas direcciones simultáneamente o en forma alternada.

2.1 Enlace de Datos

La fig. 6 indica gráficamente el enlace de datos.

Cada nodo se comunica con el otro sobre un medio de transmisión (cable telefónico, fibra óptica, enlace satelital o combinaciones de ellas), para ello requiere de una estación de enlace y de un DCE (Data Circuit Terminating Equipment).

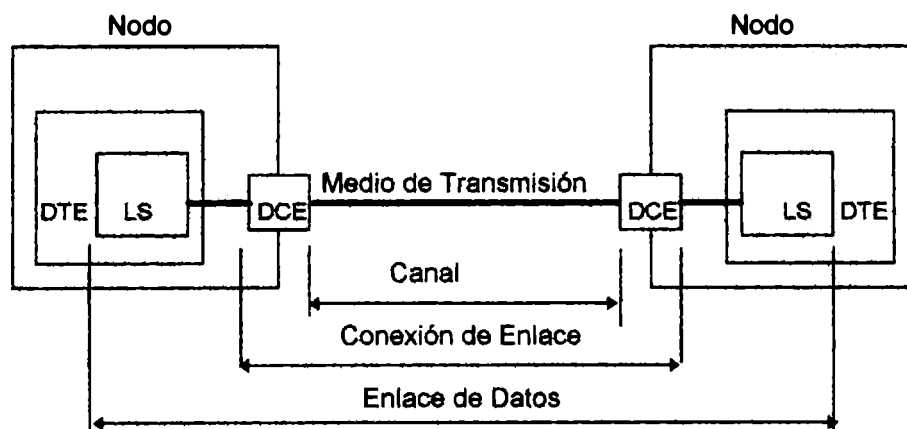


Fig.6.- Componentes de un Enlace de Datos

La estación de enlace o LS (Link Station) consiste en el hardware y el software necesarios para unir los nodos y proveer el control para el enlace. La misma es parte del DTE (Data Terminal Equipment).

El DCE establece, mantiene y cancela una conexión además de proveer la modulación necesaria de la señal para ser transmitida y recibida.

La conexión de enlace o circuito de datos incluye ambos DCEs y el canal que los comunica.

El enlace de datos consiste de:

- estaciones de enlace de los nodos que conecta,
- DCEs asociados a las estaciones de enlace,
- el canal que conecta a los DCEs.

2.2 Configuración

La fig. 7 ilustra gráficamente la configuración.

La conexión de enlace permite la configuración punto a punto orientada a conexión, con estaciones combinadas y soporta un enlace de tipo balanceado.

El flujo de información entre ambas estaciones admite que la transmisión sea half-duplex o full-duplex.

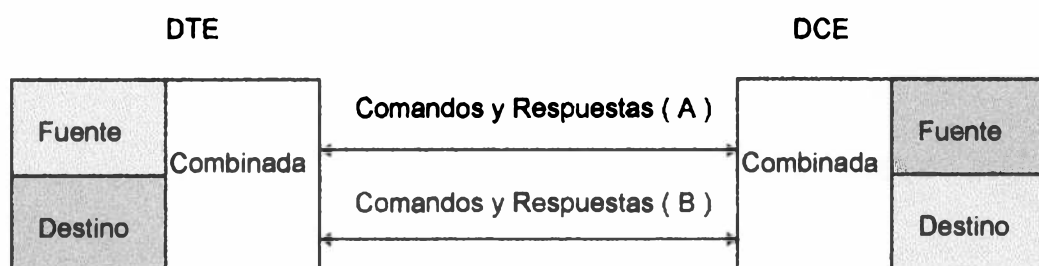


Fig. 7.- Configuración Balanceada

2.3 Gestión de Enlace

La gestión de enlace consiste en el intercambio de distintos tipos de tramas que se clasifican en: de información, no-numeradas, de control y pedidos de retransmisión.

Las tramas de información contienen los datos del usuario. Las no-numeradas se usan para funciones de control. No tienen número de secuencia y no alteran el flujo de las tramas de información. Las de control se utilizan para supervisar el flujo y los errores. Las últimas soportan el pedido como lo indica su nombre.

El establecimiento del enlace puede ser ejecutado por cualquiera de los dos nodos. Para ello cuenta con un número de comandos y respuestas que permiten efectuar la operación de conexión. Una vez establecido el enlace comienza la fase de transferencia de información que consiste en el intercambio de tramas de información y de control. Cualquiera de los dos nodos está capacitado para terminar la fase de transferencia de información enviando un comando de desconexión.

2.3.1 Establecimiento del Enlace

Debido a que el protocolo está orientado a conexión, se hace necesario como primera fase en toda transmisión, el establecimiento del enlace que consiste en el intercambio de ciertas tramas no numeradas.

La duplicidad de estas tramas no trae consecuencias inherentes al normal funcionamiento del enlace por lo cual no llevan número de secuencia.

Antes del establecimiento de la conexión de enlace, cualquiera de los nodos puede iniciar la desconexión para asegurarse que ambos están en la misma fase.

Un nodo puede querer testear el medio y, a tal fin, envía una trama. La no distinción del sentido en el cual viaja hace que sea aceptada por el receptor. El loop back, si bien la mayoría de las veces es intencional, puede ocurrir en forma accidental, de este modo tanto emisor como receptor no se darían cuenta de lo que realmente sucede. Ambos nodos podrían aceptar la trama en cuestión, el protocolo en este caso fallaría. Para evitar este problema se utiliza un campo en el diseño de trama que posibilita diferenciar entre comandos y respuestas. A causa de la configuración balanceada ambos nodos pueden enviar comandos y respuestas.

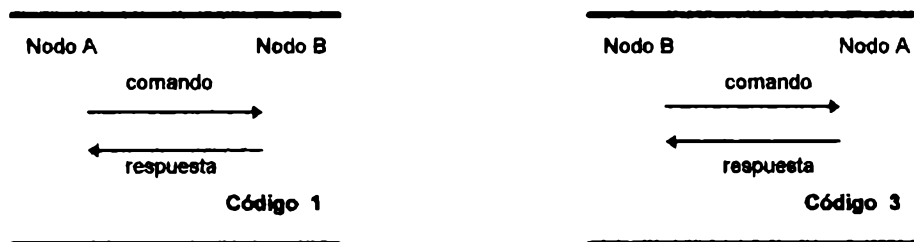


Fig. 8 - Comandos y respuestas

Si bien el campo de control indica el tipo de trama, con la distinción entre comandos y respuestas permite indicar el flujo de información en direcciones opuestas y además detectar los loop back.

Dados dos nodos A y B se distinguen los comandos de A a B con el código 1 y las respuestas de B a A con el mismo número. Mientras que se codifican con 3 los comandos de B a A y las respuestas de A a B. Según se muestra en la fig. 8.

El emisor inicia el enlace transmitiendo SABM (Set Asynchronous Balanced Mode). El receptor al aceptar contesta con una trama UA (Unnumbered Acknowledge) y pone sus variables de estado $V(R)$, $V(S)$, $V(AR)$ y el Arreglo de Retransmisión en cero. Del mismo modo cuando el UA llega al emisor se setean las mismas variables pertenecientes a ese nodo como lo muestra la fig. 9.

No obstante, tanto emisor como receptor mantienen sus propias variables de estado que determinan en que instancia se encuentra cada uno de ellos.

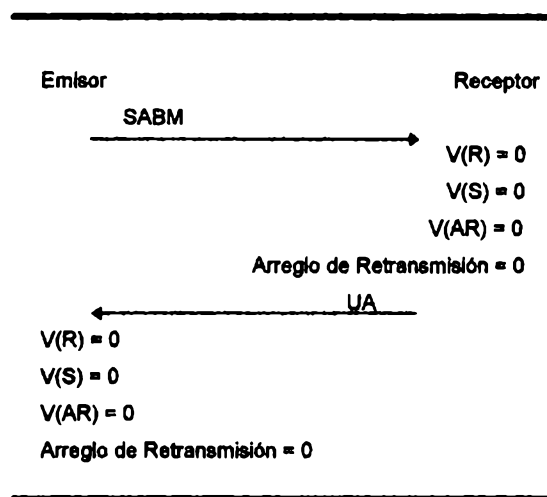


Fig. 9 - Establecimiento del enlace

- $V(S)$ variable de estado emisora: señala el número de secuencia de la próxima trama a ser transmitida, es decir $V(S) \rightarrow N(S)$ de tramas de información salientes; $V(S) := \text{módulo } N(V(S) + 1)$.
- $V(R)$ variable de estado receptora: indica el número de secuencia de la próxima trama esperada, es decir $V(R) \rightarrow N(R)$ de tramas de información y supervisoras. $V(R)$ es chequeada contra el campo $N(S)$ de las tramas de información entrantes para asegurarse que están en secuencia. Cuando arriba una trama fuera de secuencia se produce un estancamiento en esta variable, impidiendo que esta se modifique mientras no llegue la esperada.

- **V(AR)** variable de estado auxiliar receptora: indica el número de secuencia últimamente recibido fuera de orden. Es una de las variables claves que van a permitir ejecutar los mecanismos de pedidos de retransmisión.
- **VAP(R)** variable de estado auxiliar receptora de pedido: especifica el número de secuencia de la primera trama que deberá ser solicitada al emisor ante un procedimiento anormal en la transmisión.
- **Arreglo de Retransmisión:** es el eje principal que señala fehacientemente todos los estados posibles en que pueden llegar a estar cada una de las tramas ante el procedimiento de pedido de retransmisión. Como tal en la propuesta I se cuenta con cuatro estados:
 - 0: confirmación,
 - 1: pedido de retransmisión,
 - 2: aceptación,
 - 3: llegada de trama

En la propuesta II al no contar con las tramas ACC, los estados se determinan de la siguiente manera:

- 0: llegada
- 1: pedido de retransmisión.

La propuesta I cuenta con dos variables de estado adicionales:

- **SQE:** se menciona en el emisor. La misma señala la secuencia de tramas completa a la que pertenece la trama a ser transmitida, es decir $SQE \rightarrow SEQ$ para las tramas de información salientes. Los valores posibles para la misma son 0 y 1, ya que una vez que se termina de transmitir una secuencia completa, se modifica el valor de SQE a su complemento.
- **SQR:** secuenciación en el receptor. Indica la secuencia de tramas completa a la que pertenece la trama recibida. Es decir una vez que se recibe una trama se constata el bit SEQ con SQR para asegurarse que la misma pertenece al grupo esperado. En forma similar a la variable SQE, puede tomar los valores 0 y 1.

Si el receptor determina que no puede pasar a la fase de transferencia de información envía un DM al emisor negándose al establecimiento del enlace.

2.3.2 Transferencia de información

Una vez establecido el enlace comienza la fase de transferencia de información, que consiste en el intercambio de tramas de información y supervisoras y, eventualmente, de pedidos de retransmisión.

Las primeras contienen paquetes de nivel 3 incluyendo el header de esa capa dentro del campo de datos. Las segundas se usan para realizar funciones como la de confirmar tramas de información o frenar el envío. Las tramas de pedido de retransmisión son enviadas con la finalidad de que se vuelvan a mandar ciertas tramas de información.

El hecho de tener tramas de información numeradas, permite efectuar el control sobre ellas secuenciándolas. Las otras numeraciones posibilitan identificar aquellas tramas a las que se les debe aplicar determinada acción.

En base a lo ya expresado, se ahondarán en detalles para determinar un protocolo que optimice la retransmisión selectiva que permita una mejor utilización del ancho de banda.

La búsqueda de tales objetivos condujo a considerar dos propuestas de protocolos, que si bien cumplen con los requerimientos anteriores varían notoriamente, hecho que se relatará a continuación con una comparación realizada oportunamente.

2.3.2.1 Propuesta I

2.3.2.1.1 Tabla de tramas

Flag	Address	R	Control	Datos	CRC	Flag
01111110	XXXXXXX	X	XXXXXXXX	XXXXXXXX >= 0 (múltiplo de 8)	XXXXXXXXXXXXXXXXXX	01111110

Fig. 10 - Formato de trama

Campos	R	de Control
Supervisoras	0	10 Código SEQ N(R)
• RR	0	10 00 X XXX
• RNR	0	10 10 X XXX
• ACC	0	10 01 X XXX
Información	0	0 N(S) SEQ N(R)

Fig. 11 - Campos de Retransmisión y de Control

No Numeradas	0	11	Código
• SABM	0	11	111100
• UA	0	11	001110
• DM	0	11	111000
• DISC	0	11	001010
• FRMR	0	11	101001
Pedido de Retransmisión	1	0	N(S) SEQ N(R)

(*) X = valor de un bit

Fig. 11 - Campos de Retransmisión y de Control (Continuación)

2.3.2.1.2 Desarrollo

En virtud de la complejidad que denota la descripción completa del caso real de la transferencia de datos, se considerará en primera instancia el caso en que el emisor envíe tramas al receptor y éste las recibe del mismo modo en que le fueron transmitidas, es decir, en orden y sin errores. El canal no pierde tramas ni introduce ruido.

La Capa de Enlace del nodo emisor recibe paquetes de la Capa de Red. A cada uno de ellos le agrega un header que será analizado por la capa de enlace del nodo receptor. Uno de los elementos que se destacan en el header es el número de secuencia de la trama que se obtiene a partir de la variable V(S).

Una vez constituida la trama de información se guarda en el buffer. Un parámetro de implementación dependiente de este espacio disponible para almacenar las tramas es la ventana. Se denomina de este modo al número máximo de tramas pendientes de confirmación permitidas.

Luego se envía al receptor a través de la Capa Física y se incrementa la variable V(S) en una unidad.

En el nodo receptor, la Capa de Enlace se encarga de tomar de la Capa Física una secuencia de bits para ser analizado. Cuando se reconoce la trama se verifica su número de secuencia con la variable de estado V(R), y se almacena en el buffer de recepción, incrementándose en 1 el límite superior de la ventana. Luego V(R) también aumenta en 1 su valor. En algún instante antes de que se agote la ventana envía en bloque el grupo de tramas de

información recibidas, es decir la secuencia de tramas ordenada y completa es remitida a la Capa de Red y modifica su ventana usando la técnica de ventana deslizante.

De inmediato manda al emisor la confirmación correspondiente a este grupo. La misma puede ser efectuada de dos modos:

- utilizando la técnica de piggybacking ó
- con RR en caso de no existir tramas de información a ser enviadas de dicho nodo al emisor.

El número de secuencia señalado como $N(R)$ o dentro de RR será provisto por el $V(R)$ local.

Cuando al emisor le llega la confirmación realiza el correspondiente corrimiento de ventana, vaciando el buffer de acuerdo a las tramas asentidas. De este modo permite el almacenamiento de nuevas tramas a partir de paquetes de la Capa de Red que luego serán enviadas al receptor.

Lo anteriormente descrito está enmarcado en un medio físico libre de error. No obstante, un medio real está afectado a distintos fenómenos que pueden perjudicar la integridad de una trama. En consecuencia, esto conduce a la efectivización de ciertos mecanismos entre emisor y receptor, que posibiliten una transmisión que recupere fallas de un modo eficiente.

Debido a que el ruido puede afectar la transmisión, cuando la Capa de Enlace del nodo emisor recibe un paquete de la Capa de Red, no sólo le incorpora un header, sino que también le calcula el CRC correspondiente. Este es anexado a la trama constituyendo el trailer de la misma. Ya en el receptor, la Capa de Enlace verifica el estado de la trama y se determina si su número de secuencia módulo $(n - 1)$ está comprendido dentro de los límites de la ventana. En caso contrario se la descarta.

En un canal ruidoso no sólo se dañan tramas sino también pueden llegar a perderse. Para salvar este inconveniente, se asocia a cada trama en el emisor, un temporizador con la finalidad de que la misma se retransmita luego de un cierto lapso de tiempo, si no hubo asentimiento de parte del receptor.

El receptor está obligado a asentir las tramas que llegaron correctamente mediante RR o usando la técnica de piggybacking. A causa de la gran variedad de tareas que efectúa el receptor, el procesamiento de tramas almacenadas en el buffer puede ser diferido a tal punto que cause el vencimiento en cadena de los temporizadores asociados a las mismas en el emisor. Para evitar esta situación, el receptor envía un $RNR(n)$, donde n indica la confirmación de hasta la trama $(n-1)$ y además frena el envío de nuevas tramas por parte del emisor. Este estado se mantiene hasta que el receptor envíe un RR o un pedido de retransmisión con los números asociados a dichas tramas de control, activando de este modo la emisión de tramas.

Es posible que el receptor demore los asentimientos hasta tanto la Capa de Red le envíe un paquete. Como consecuencia, el emisor puede comenzar a retransmitir tramas que el receptor recibió correctamente debido al vencimiento de los temporizadores vinculados a cada una de ellas. Este hecho motiva una mala utilización del canal. Por otra parte un receptor que envíe

confirmaciones de pequeños grupos de tramas llevaría a los mismos efectos ya que abrumba el canal con asentimientos que podrían llegar a ser redundantes provocando además un tiempo extra en el procesamiento de los mismos en el emisor. Por ejemplo si se envían seguidamente RR(3) y RR(4), este último confirmaría las anteriores sin necesidad de haberse transmitido RR(3).

Por consiguiente, con la finalidad de obtener una mayor performance del canal y del ancho de banda se utiliza un temporizador en el receptor. Su vencimiento se halla en coordinación con los tiempos del temporizador asociado a la primer trama del grupo emitido. Con la llegada de la primera trama de información se pone en marcha el temporizador. Por cada confirmación efectuada se resetea el timer y se vuelve a inicializar con el arribo de la próxima trama de información.

Un inconveniente que puede llegar a ocasionar esta optimización, se presenta cuando el canal es demasiado ruidoso, con lo que el RR podría llegar a perderse o a no ser reconocido. En tal caso el emisor retransmitiría aquellas tramas que en el receptor fueron confirmadas. El receptor al enterarse que se trata de tramas recibidas con anterioridad reitera su RR.

Un hecho relevante en el diseño de este protocolo consiste en la ampliación del tamaño máximo de ventana. Con un número de secuencia módulo n se permiten $n - 1$ tramas sin confirmar, lo que indica una ventana de tamaño máximo $n - 1$. En otros protocolos la ventana máxima permitida era restringida a $(n - 1)/2$. La diferenciación de las secuencias posibilita ampliar el rango de la ventana. Una secuencia se distingue con la siguiente alternando el valor correspondiente del bit SEQ. El campo SEQ tiene como longitud un bit y forma parte del campo de control de las tramas de información, de las tramas de control numeradas y de las de pedido de retransmisión. Por ejemplo, con números de secuencia de 0 a 7 se tiene una ventana máxima de tamaño 7. Supongamos que el emisor envía las tramas 0, 1, ..., 6; éstas se transmiten con el bit SEQ en 0. El receptor al recibirlas las confirma modificando su ventana. Al emisor le llega el asentimiento, por lo tanto éste se dispone a enviar las próximas, es decir la siguiente ventana. Las tramas son 7, 0, ..., 5. La 7 al pertenecer a la secuencia anterior se envía con el bit SEQ en 0, mientras que las restantes se transmiten con el mismo bit en 1 ya que pertenecen a la siguiente secuencia.

Como se mencionó anteriormente luego de reconocida la trama se constata su integridad tratando de detectar errores de CRC. Posteriormente se testea si se trata realmente de un tipo de trama definido, en este caso de información. Luego se verifica si su número de secuencia se halla dentro del rango de la ventana. De este modo pueden darse distintas situaciones. Si el número de secuencia se halla fuera del rango de la ventana actual, se descarta esa trama reiterando al emisor la última confirmación.

Cuando está dentro de las cotas inferior y superior, se verifica la pertenencia a la secuencia constatando el bit SEQ, y además si corresponde en orden a la esperada entonces se la guarda en el buffer.

Si bien la trama cae en la ventana puede suceder que se halle fuera de secuencia. En tal caso se deben seguir una serie de pasos. En primera instancia se guarda la trama recibida en el buffer y se verifica la existencia de aquellas almacenadas que llegaron anteriormente en secuencia. Estas son confirmadas siguiendo el curso correspondiente. Aquellas tramas faltantes comprendidas hasta la que llegó en última instancia necesitan que sean enviadas nuevamente. Para ello primero se las señala en el arreglo de retransmisiones. Luego se efectúa el pedido de retransmisión con la correspondiente trama de control, indicando en la misma el rango de las tramas ausentes.

Consideremos el siguiente ejemplo, para ello usaremos el formato señalado por la fig. 12 para la referenciación de las tramas.

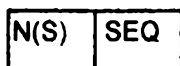
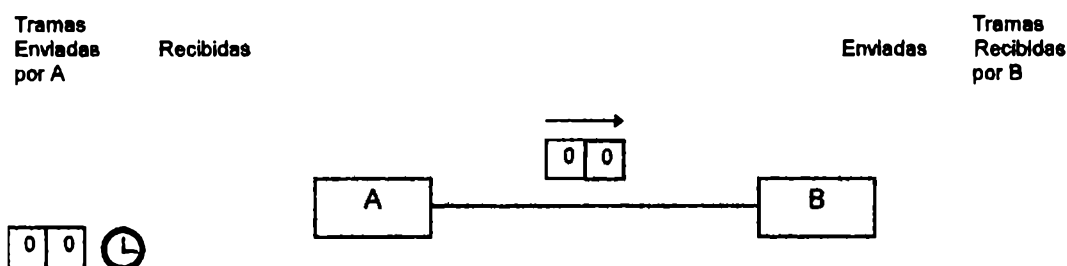


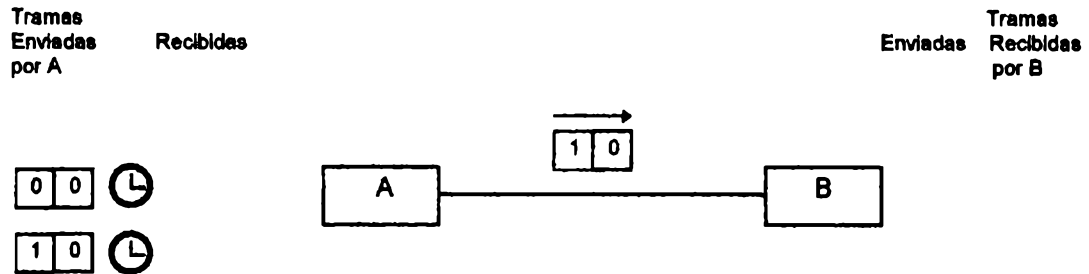
Fig. 12 - Trama de Referencia

1. La Capa de Red del emisor le envía paquetes a la Capa de Enlace. Se almacenan los paquetes en el buffer de emisión y como se trata de tramas de información, se le deben agregar los campos identificatorios.
2. SQE mantiene el número de secuencia completa a la que pertenece la trama a transmitirse. El valor de SEQ es asignado desde SQE. Luego de la asignación se verifica si la próxima trama que se envía pertenece a su grupo de secuencia o a la siguiente. En base a ello se modifica o no el valor de SQE. El N(S) y el N(R) se obtienen de las variables de estado V(S) y V(R) respectivamente. Luego de la asignación V(S) aumenta en uno su valor. Se completan los demás campos necesarios para conformar la trama de información y luego se calcula el CRC, incorporándose al final de la misma. Para enviar la primer trama se obtienen los valores necesarios para estructurar la misma: SQE con 0 se asigna a SEQ, N(S) en 0 se logra a partir de V(S) y N(R) con 0 señala que no hubo tramas de información por parte del receptor. Se constatan los valores de SQE y V(S), en el caso de ésta última además se la incrementa en 1. Una vez estructurada la trama se la transmite y se pone en marcha el temporizador vinculado a

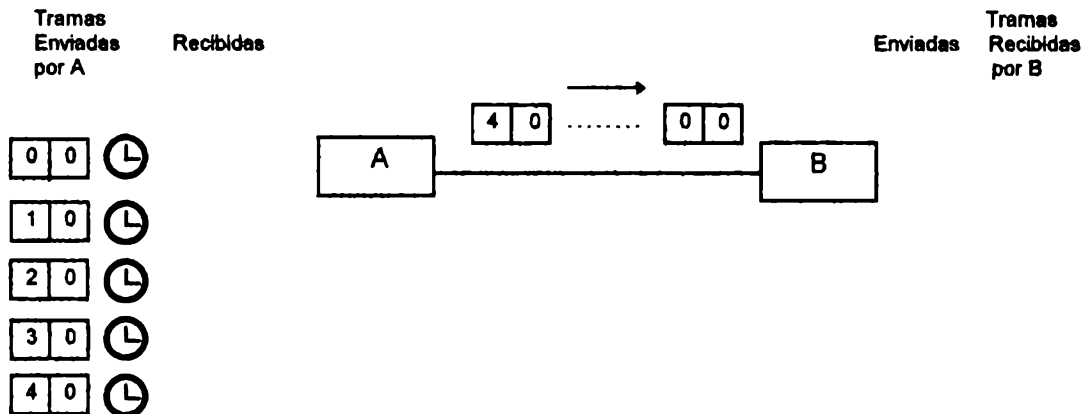


la misma .

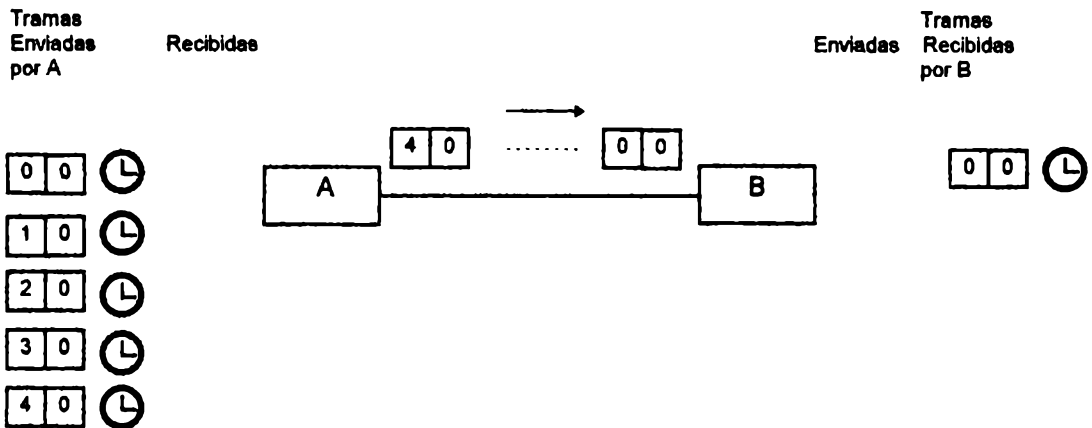
- Para la próxima trama se toman estos valores de SQE y de V(S) para asignarlos a SEQ y N(S) respectivamente. N(R) sigue en su estado original. Si bien SEQ sigue valiendo 0 porque no se termina de enviar la secuencia, V(S) ahora toma el valor 2. Construida de ésta manera se transmite la trama 1, poniéndose su temporizador en marcha.



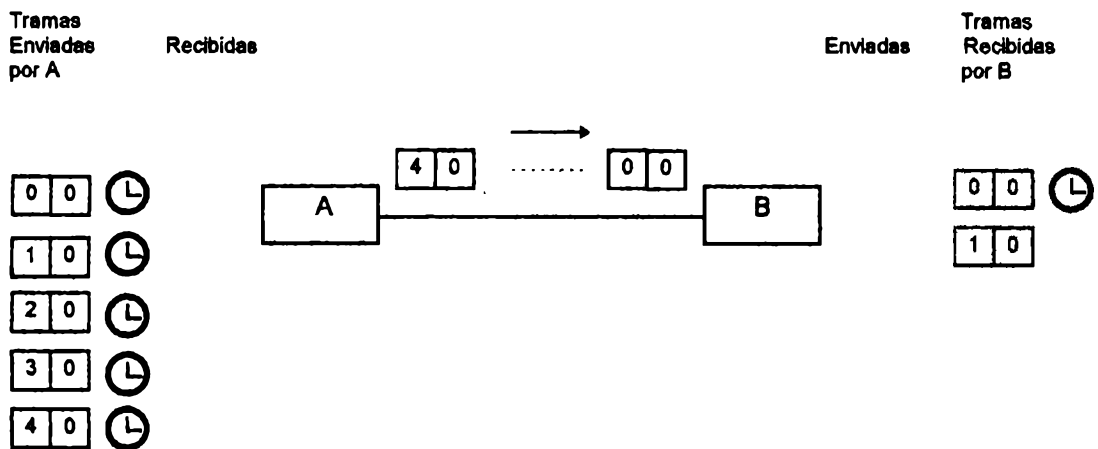
- Del mismo modo para todas las tramas de información que necesitan enviar se usa el mismo mecanismo con las variables de estado.
- Se envían las tramas 2, 3 y 4, con ello se inicializan los temporizadores correspondientes.



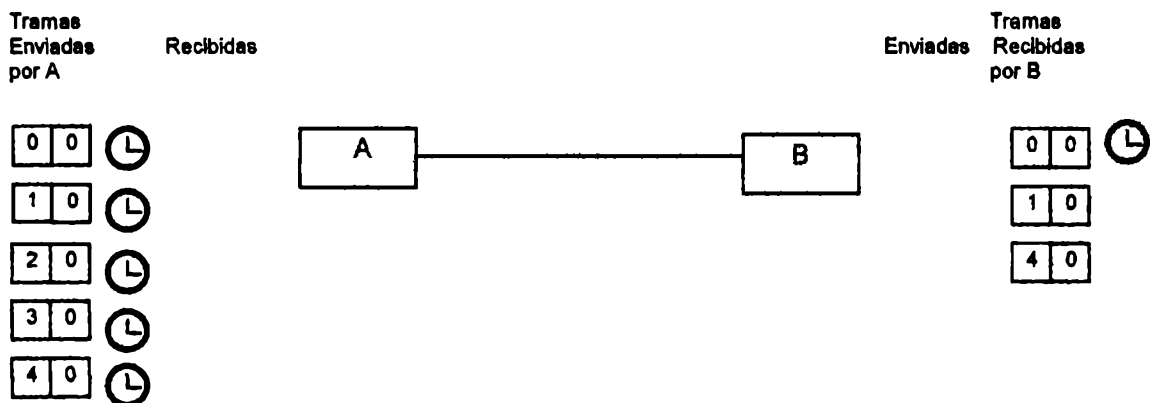
- Una vez que el receptor reconoce una trama la analiza y si no detecta error, determina si su N(S) cae dentro de la ventana y además verifica que el campo SEQ tenga el valor 0 ya que así pertenece a la primer secuencia. Al recibir la trama 0 se constata si coincide con el V(R) local, si son iguales, se aumenta en 1 dicha variable de estado, es decir $V(R)=1$. Además por ser la primera trama recibida se pone en marcha el temporizador de recepción y se almacena en el buffer el paquete correspondiente a la trama.



7. Luego se reciben las tramas cuyo N(S) es 1 con SEQ concordante con dicha secuencia; como el número de secuencia se corresponde con el V(R) se la incorpora al buffer, y V(R) adquiere el valor 2.

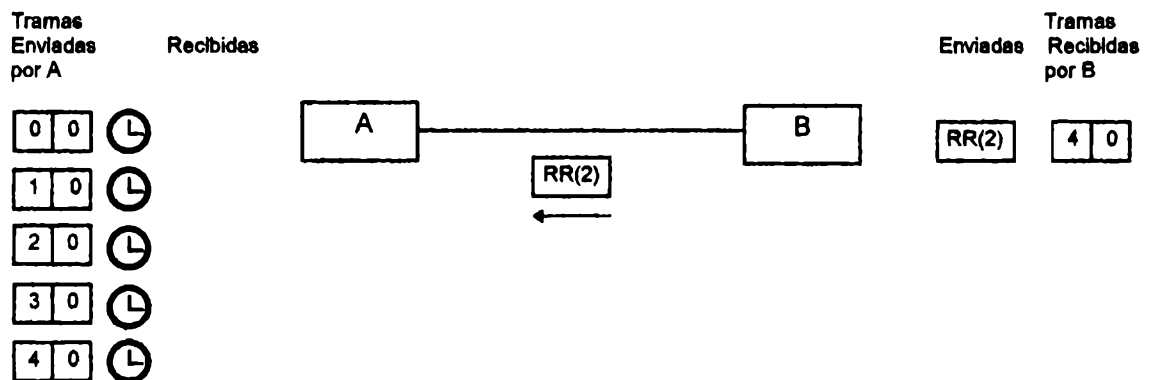


8. Seguidamente se reconoce una nueva trama, donde el N(S) es 4. Si bien cae en la ventana, no es consecutivo al anterior, con lo cual se trata de una trama fuera de secuencia. Debido a

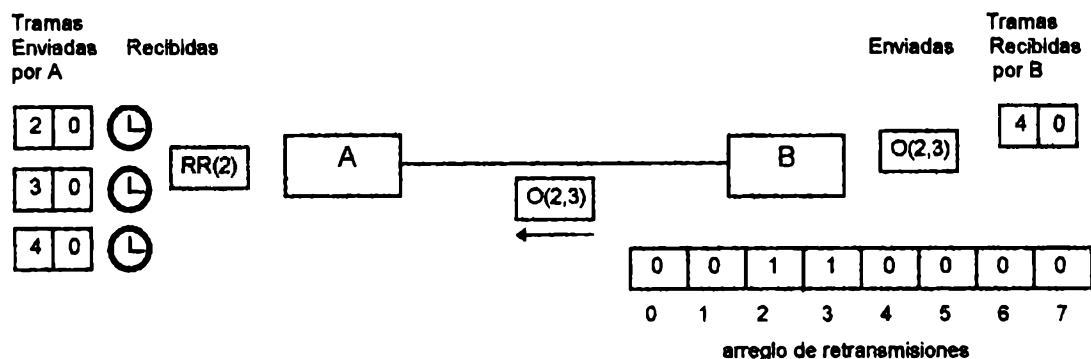


que el protocolo admite esta situación, guarda la trama dentro del buffer, la variable de estado $V(AR) = 4$ denota que es la última trama llegada no en secuencia.

9. Las tramas 0 y 1 siguen un orden de numeración, por lo tanto son confirmadas. Como se mencionó anteriormente, puede ser realizado mediante un RR o a través de una trama de información. En ambos casos, el número de secuencia que acompaña al asentimiento se obtiene de $V(R)$. Se desactiva el temporizador de recepción y se envían sendas tramas a la capa superior, efectuando el correspondiente corrimiento de ventana. Por otro lado se desechan los paquetes de su buffer.

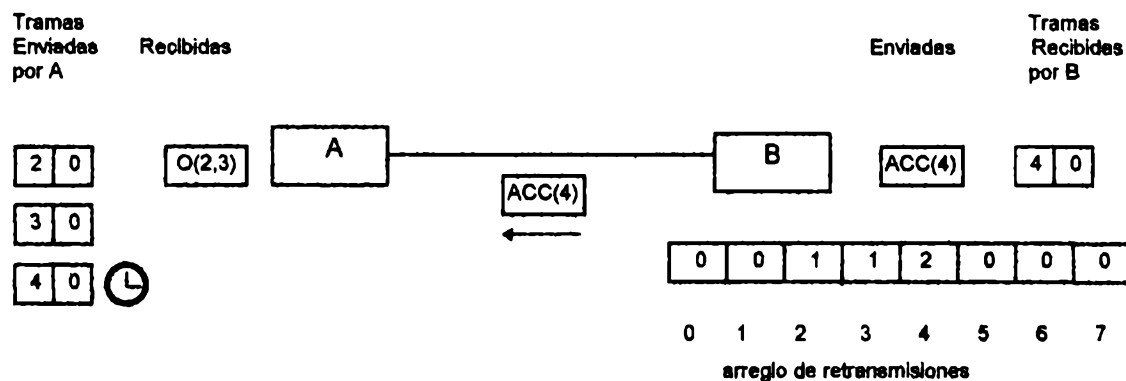


10. El arribo de la confirmación de las tramas 0 y 1 al emisor, hace que éste realice el corrimiento de ventana además de eliminar sus paquetes del buffer. Simultáneamente se desactivan los temporizadores asociados.
11. Se indica en el arreglo de retransmisiones las tramas que serán pedidas al emisor para que se vuelvan a enviar. El pedido de retransmisión siempre se origina debido a que una o más tramas no se han detectado como tales. Es decir el ruido existente en el medio dañó o en el peor de los casos perdió tramas. En el ejemplo se incorporan las $N(S)$ 2 y 3, con sus estados respectivos. El estado de pedido de retransmisión es 1.
12. Se estructura la trama con el pedido de retransmisión donde al $N(S)$ le corresponde el $N(S)$ de la primera trama ausente, y al $N(S+1)$ el $N(S)$ de la última. Para el caso que se trata son $N(S) = 2$ y $N(S+1) = 3$. Es lógico pensar que si se tratase de una única trama, ambos $N(S)$ y $N(S+1)$

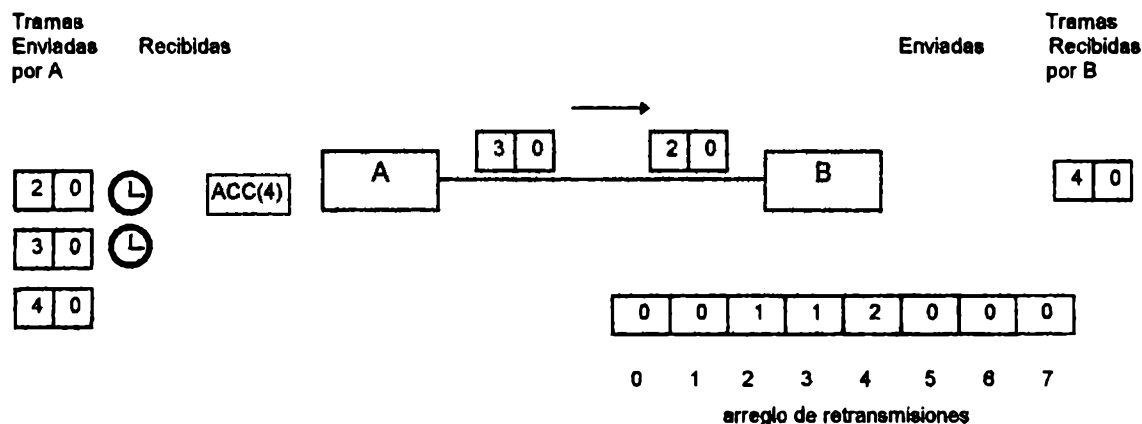


responderían a ese número de secuencia. Esta trama debe ser enviada al emisor.

13. Las tramas posteriores a las pedidas deben ser aceptadas. El proceso de aceptación consiste en guardar el N(S) en el arreglo de retransmisiones en el lugar correspondiente con el estado en 2, es decir señalando que es una aceptación. Luego se arma la trama ACC con N(R) igual a dicho N(S). Para el ejemplo el 4 se almacena en el arreglo con código 2.

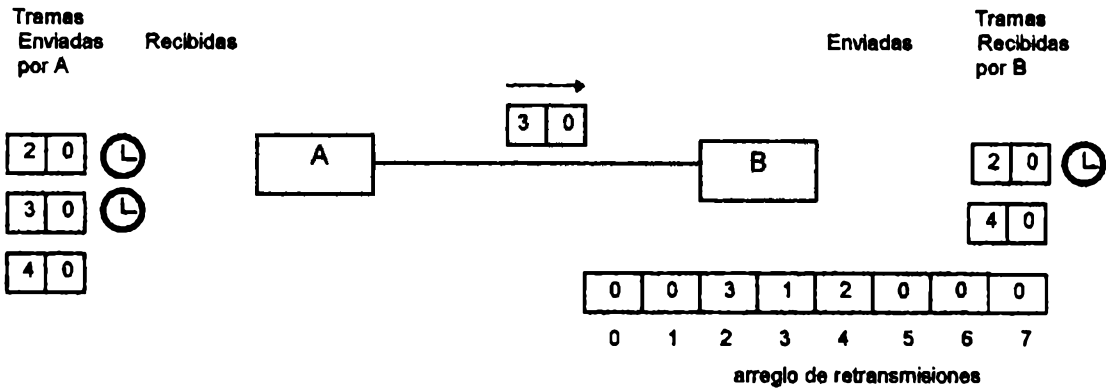


14. La llegada de una trama ACC al emisor tiene como finalidad desactivar el temporizador asociado a la trama en cuestión. Es decir al arribo de ACC(4) se desactiva el temporizador de esa trama.
15. El emisor al arribo de la trama del pedido de retransmisión analiza los N(S) y N(S+1), disponiéndose a enviarlas nuevamente inicializando otra vez los temporizadores correspondientes en el momento del envío. En este caso vuelve a enviar las tramas 2 y 3.

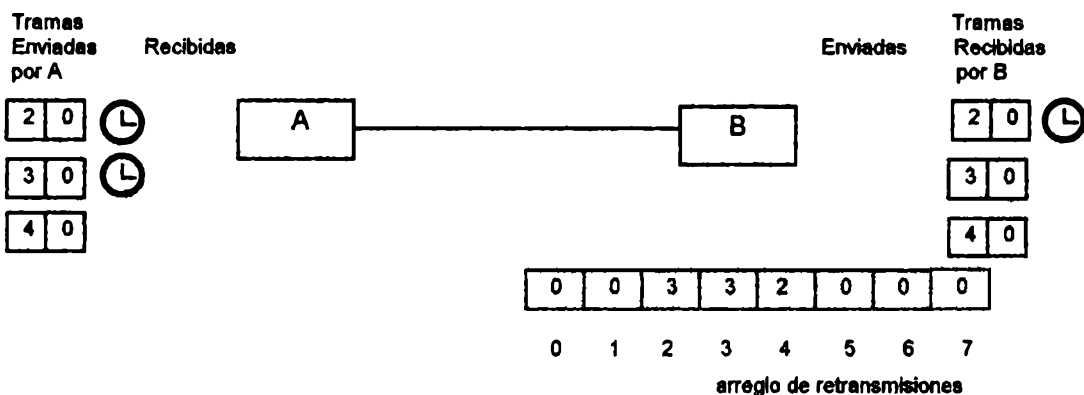


16. En el receptor se reconoce la trama retransmitida con N(S) = 2. Al verificarse en el arreglo que es una trama pedida con anterioridad y es la primera del rango pedido se pone en marcha el temporizador de recepción. Además se modifica el estado de la trama a 3, indicando de la llegada de la misma. Posteriormente se debe testear cual es el estado de la trama subsiguiente en el arreglo de retransmisiones. En este caso el estado de la trama 3 se halla en 1, por lo tanto sigue quedando en espera del arribo de la retransmisión. Cabe destacar que si en lugar

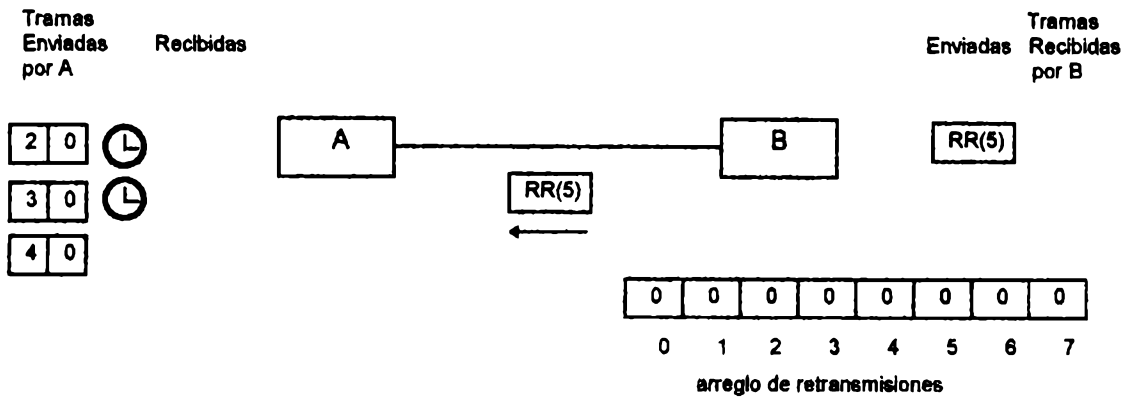
de haber llegado la 2 se presenta la 3, por ser retransmitida se constata en el arreglo si hubo un pedido de la misma. Al no tratarse de la primera trama solicitada, ésta debe aceptarse según el proceso de aceptación correspondiente, desencadenando un nuevo pedido de retransmisión de la trama 2.



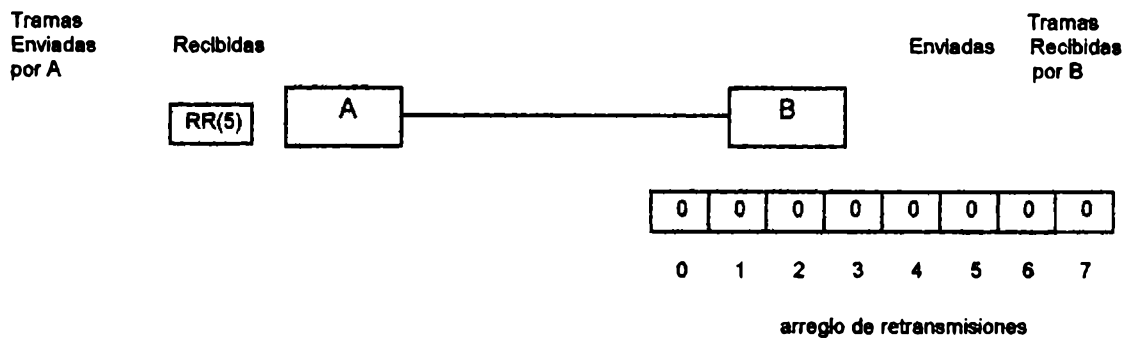
17. El tratamiento a realizar a la llegada de la trama con $N(S) = 3$ es similar al de la trama 2. No obstante, no se pone en marcha el temporizador de recepción debido a que todavía no hubo vencimiento del mismo.



18. Se efectúa nuevamente la verificación del estado de la próxima trama en el arreglo. La trama 4 tiene asignado el estado en 2, como se mencionó, ésta fue aceptada. Debido a que 4 es el $N(S)$ de la trama aceptada en secuencia más alta, la variable de estado $V(R)$ se modificará al valor del $N(S) + 1$. Se desactiva el temporizador y se genera el proceso de asentimiento de estas 3 tramas, es decir 2, 3 y 4. Por otra parte se resetean los estados correspondientes a estas tramas, como a ellas mismas.



19. Se envían las tramas asentidas a la Capa de Red.



Si llega al receptor una trama que no presenta error de CRC, pero que al querer desglosarla se detectan errores en su sintaxis, se descarta la misma y se desencadenan todos los mecanismos que conducen a la emisión del FRMR, indicando con ello que se halla ante un error irre recuperable. Posteriormente se envía un DISC.

El hecho de retransmitir cada trama por vencimiento del temporizador en el emisor lleva asociado un contador que permite un número máximo de retransmisiones, alcanzado ese valor, el emisor envía un DISC con la finalidad de indicarle al receptor que necesita desconectarse. Acto seguido cualquiera de los dos extremos podrá reinicializar el enlace.

Cuando la Capa de Red permanece por mucho tiempo inactiva o se transmitieron todas las tramas de información posibles de la ventana, el emisor envía sendos RR a intervalos de tiempo determinados. Del mismo modo se usa una política similar cuando el receptor no tiene más tramas para enviar.

La trama RNR se utiliza con el propósito de señalar ciertos problemas que se presentan temporalmente en el receptor, como por ejemplo: la escasez de memoria temporal. Si el buffer del receptor está lleno y éste no logra procesar las tramas pendientes, envía un RNR al emisor para que frene el envío de las tramas.

2.3.2.2 Propuesta II

2.3.2.2.1 Tabla de tramas

Flag	Address	R	Control	Datos	CRC	Flag
01111110	XXXXXXX	X	XXXXXXXX	XXXXXXXX	XXXXXXXXXXXXXXXXXX	01111110
≥ 0 (múltiplo de 8)						

Fig. 13 - Formato de trama

Campos	R	de Control			
No Numeradas	0	11	Código1 P/F Código2		
• SABM	0	11	1111	P	000
• UA	0	11	0011	F	110
• DM	0	11	1110	P/F	000
• DISC	0	11	0010	P	010
• FRMR	0	11	1010	F	001
Supervisoras	0	10	Código	P/F	N(R)
• RR	0	10	00	P/F	XXX
• RNR	0	10	10	P/F	XXX
• REQ	0	10	01	P/F	XXX
Información	0	0	N(S)	P	N(R)
Pedido de Retransmisión	1	0	N(S)	P/F	N(R)

(*) X = valor de un bit

Fig. 14 - Campos de Retransmisión y de Control

2.3.2.2.2 Desarrollo

Como contrapartida a la propuesta anterior se pensó en otra con características propias, sin embargo sigue cumpliendo con los requisitos planteados.

La diferencia fundamental con la propuesta I se basa en la inexistencia de temporizadores vinculados a cada trama de información en el emisor. No obstante, se mantiene el tamaño de ventana ($n - 1$), para lo cual se cuenta con un único temporizador asociado a la última trama que se va transmitiendo.

De igual modo el receptor cuenta con un temporizador que determina la confirmación de tramas.

Por un error en la transmisión, el emisor puede no recibir asentimiento de una secuencia de tramas. En tal caso, expirado su temporizador, comienza a transmitir tramas de control REQ a tiempos lógicos y constantes solicitando una respuesta inmediata del receptor, quien deberá responder inexorablemente con una trama de control. Si tuviese que enviar pedidos de retransmisión, éstas tendrán prioridad ante los RR, postergándose el envío de los mismos a la llegada de las tramas solicitadas. En el caso que el receptor esté congestionado, enviará el correspondiente RNR que le indicará al emisor que deberá esperar, posponiendo el disparo de las tramas REQ.

Enviadas ($n - 1$) tramas REQ y no habiendo recibido ninguna contestación, se procede a enviar un DISC con el propósito de desconectarse. Luego de ello, cualquiera de los dos puntos podrá volver a restablecer el enlace.

Quando la Capa de Enlace del emisor recibe un paquete de su Capa de Red estructura la trama de información agregando un header con el objetivo de que la misma pueda ser reconocida en el receptor y un trailer con el CRC correspondiente. Si bien el formato es muy similar a la propuesta anterior, un punto importante a tener en cuenta es la no utilización del bit SEQ que permitiría diferenciar secuencias de tramas. En ésta propuesta no se hace necesario su uso, ya que el control se efectúa con un sólo temporizador asociado a las tramas en el emisor, que dispare un REQ ante un desconocimiento del estado del otro extremo. Es decir no hay una retransmisión a ciegas, sino que toma una decisión luego de que se le responda el REQ. Esta acción evita un solapamiento de ventanas.

Otro tema que tiene en cuenta esta propuesta es la incorporación del bit P/F, que es usada como poll para solicitar una respuesta rápida y como final para indicar que una trama en particular es respuesta a un poll.

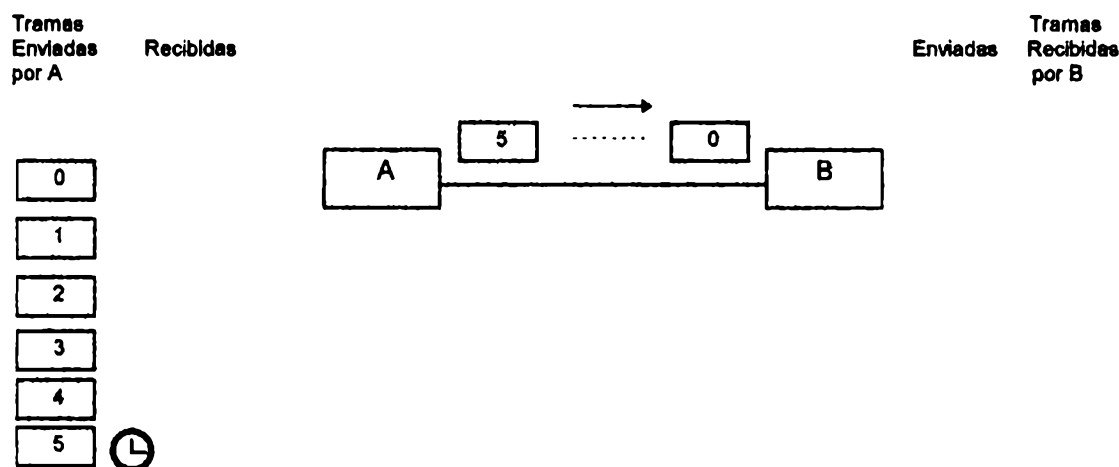


N(S)

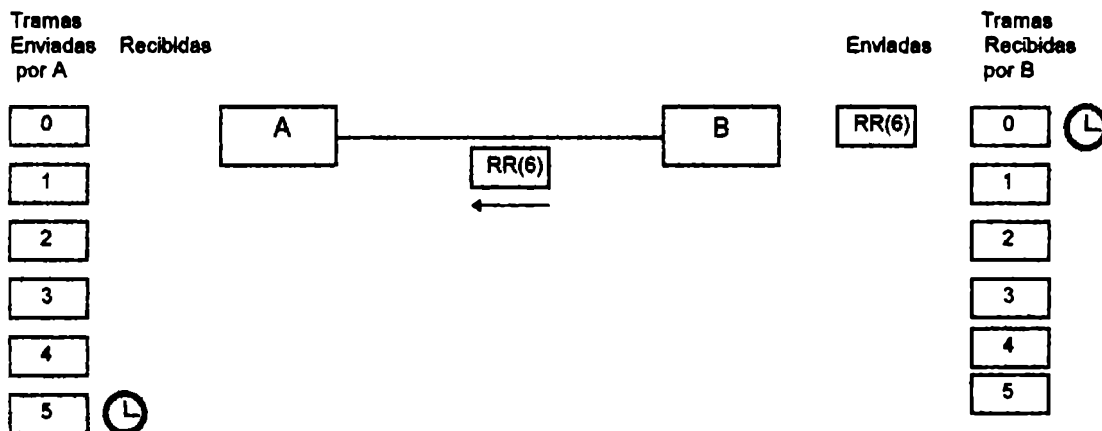
Fig. 15 - Trama de Referencia

Consideremos el siguiente ejemplo, para ello se usará el formato indicado por la fig. 15 para la referenciación de las tramas.

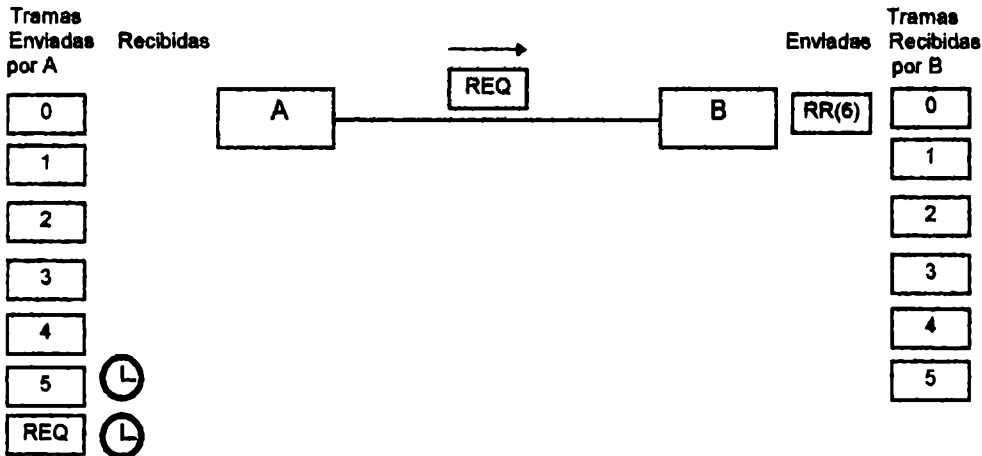
1. Cuando la Capa de Enlace del emisor envía la trama 0, se inicializa el temporizador. Por cada trama de información que transmite vuelve a reinicializarlo, es decir si se mandan con N(S) en 1, 2, 3, 4 y 5, la última puesta del temporizador se realiza cuando se transmite la 5.



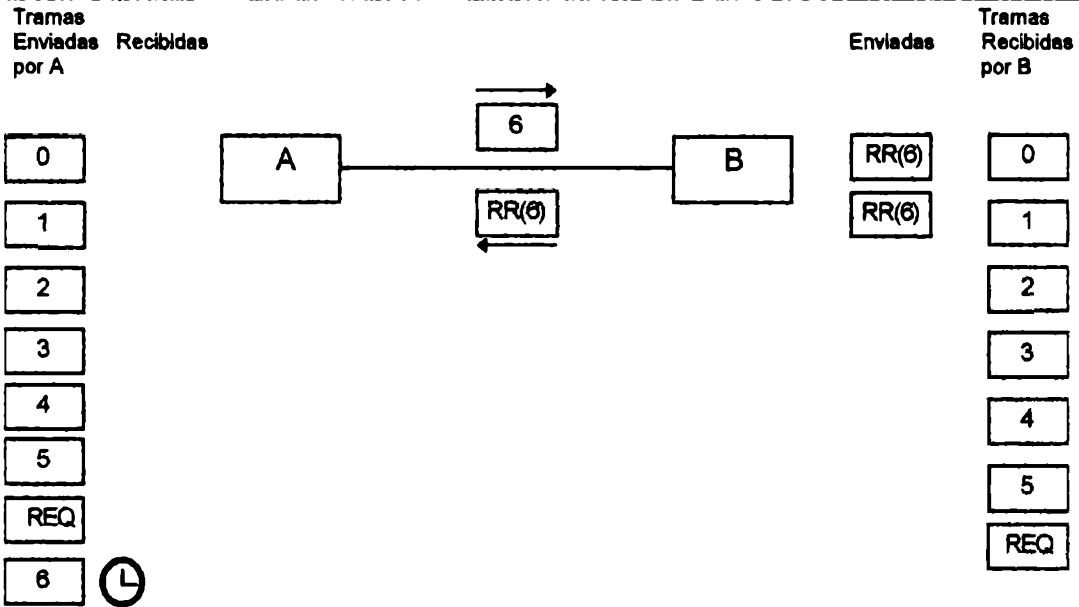
2. Al distinguirse una trama en el receptor, se analiza para ver si detecta error de CRC. Al no presentar errores y luego de reconocer la trama de información, se verifica que N(S) se encuentre en el rango correspondiente a la ventana. Cuando se reconoce la trama 0 al ser la primera perteneciente al rango de la ventana actual se pone en marcha el temporizador de recepción. Seguidamente se reciben las tramas de 1 a 5 en secuencia. En ese instante expira el temporizador, lo que impulsa al receptor a transmitir un RR con $N(R) = 6$, en el caso que no hubiese habido piggybacking. Por consiguiente, se hace el correspondiente cambio de ventana.



3. Si bien el receptor envió la confirmación en el emisor no se recibió tal trama, con lo cual expira el temporizador determinando el envío de una trama REQ.



4. Al transmitirse una trama REQ, se coloca una variable booleana VREQ en falso. Si pasado el tiempo prudencial predeterminado y no hubo contestación del otro extremo, se vuelve a emitir un nuevo REQ. En el caso de que haya habido respuesta, VREQ se modifica a verdadero, impidiendo de esta forma que se envíen más tramas REQ por el momento, y para evitar posibles malas interpretaciones del estado del otro punto.
- Por ej. el emisor envía las tramas 6, 7, 0 y 1, poniendo en marcha el timer en última instancia con la transmisión de la 1. Vencido éste, comienza a enviar REQ. Al receptor le llegan las tramas, envía RR(2) y cambia su ventana, antes del arribo del REQ. El emisor pone en línea una nueva trama REQ con anterioridad a la llegada del RR(2), es decir se envía el segundo REQ por no haber recibido respuesta del otro extremo. El primer REQ en el receptor lo obliga a emitir otro RR(2). En el emisor las dos tramas RR(2) le llegan casi simultáneamente. La interpretación que le da a esta situación el emisor es cambiar la ventana y enviar las tramas de la 2 en más con la primera confirmación. Posteriormente con la llegada de la segunda retransmitiría las mismas. En este caso el protocolo falla porque pierde el control ante los REQ enviados. Esta situación puede ser resuelta a través de la utilización de la variable VREQ.
5. Retomando a la situación original, luego de enviada la primer trama REQ y antes que le fuese contestada, la Capa de Red le transfiere otro paquete, entonces se lo rotula con $N(S) = 6$ y anexado todos los campos necesarios se la emite poniendo de nuevo el timer.
6. Al receptor le llega la trama REQ lo que lo obliga a realizar una contestación, en este caso vuelve a enviar RR(6).



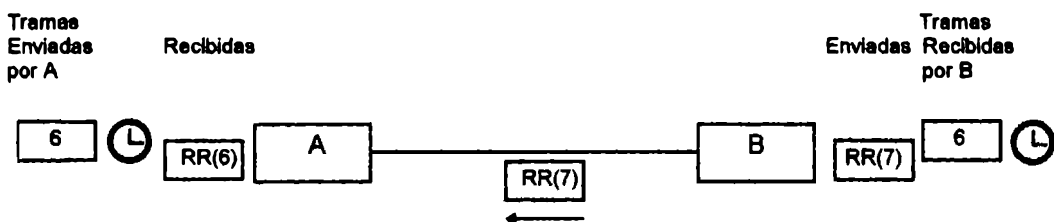
7. A continuación recibe la trama 6, guarda en el buffer el paquete, poniendo nuevamente el timer en marcha.

8. Al emisor le llega el RR(6) y coloca VREQ en verdadero. Sin embargo, dado que se emitió con posterioridad al REQ una trama de información, tomará al RR(6) como un asentimiento para las tramas anteriores a 6, modificará la ventana pero hará caso omiso a la retransmisión de la trama antedicha.

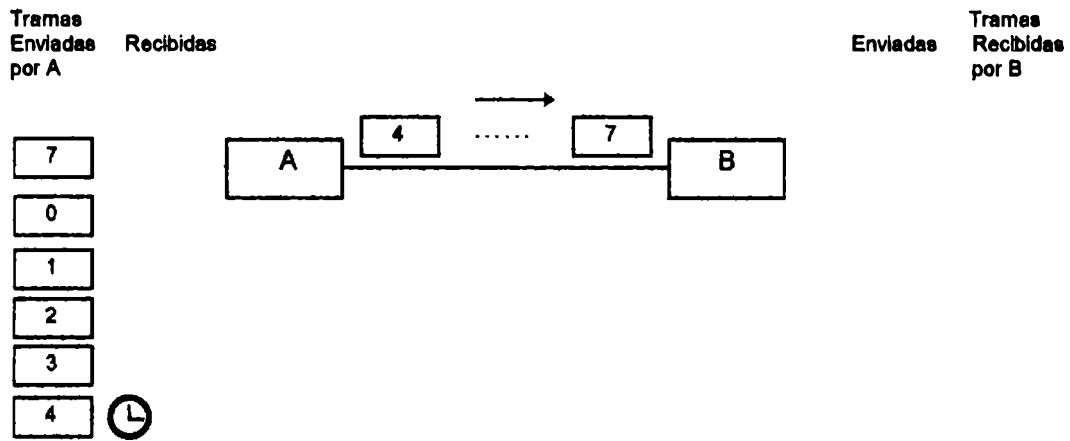
Se contempló la posibilidad de tener los dos casos frente a la llegada de un RR como contestación a un REQ.

- Cuando se carece de tramas de información transmitidas luego del REQ, tal como ocurre en la emisión de la/s última/s trama/s, si la Capa de Red retarda el envío de paquetes, o se completó la ventana de emisión, debe constatar si el RR satisface a todas las tramas enviadas, en caso contrario deberán retransmitirse aquellas que no llegaron a confirmarse.
- Si en el momento de recibir un RR ya se envió alguna trama de información posterior al REQ no se debe tomar en cuenta la funcionalidad de la retransmisión. Este caso se plantea en el ejemplo.

9. Cuando llega al receptor la trama 6, éste la confirma con RR(7) y modifica su ventana.

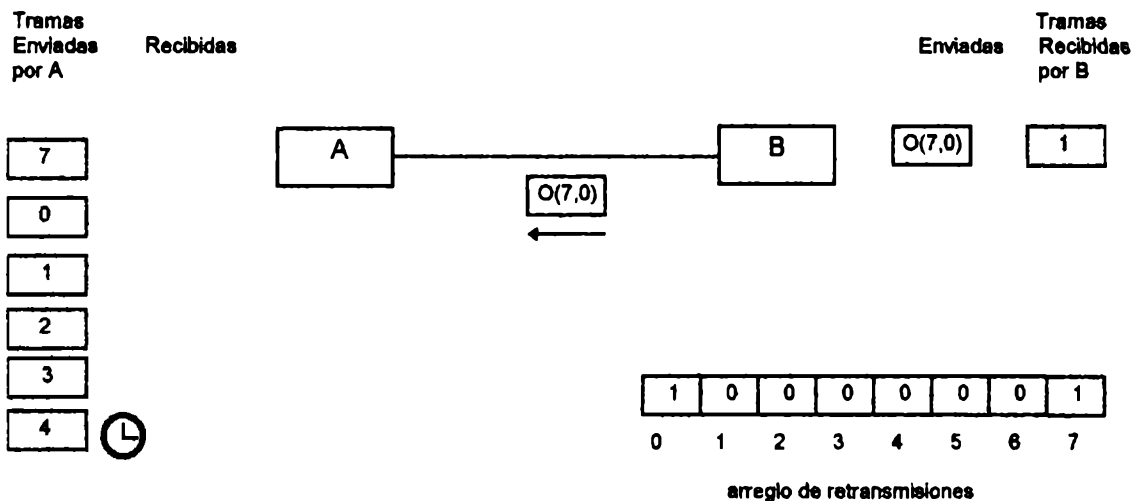


10. El RR(7) arriba al emisor, permitiéndole al mismo efectuar el correspondiente corrimiento de ventana. Consecuentemente envía las tramas subsiguientes: 7, 0, ... , 4. La última reinicialización del temporizador se efectúa con la salida de la 4.



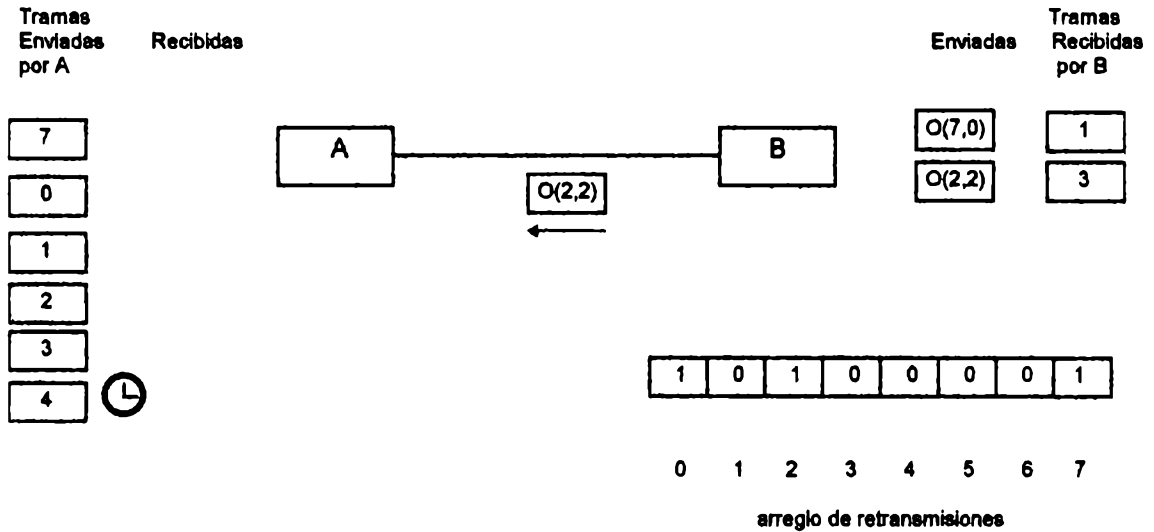
11. Al receptor le llega la trama 1. Debido a que se trata de una fuera de secuencia primeramente se la almacena en el buffer y no activa el timer de recepción. Se indica en el arreglo de retransmisiones las tramas que serán solicitadas, es decir 7 y 0 con los estados en 1.

12. Se estructura una trama de pedido de retransmisión con $N(S) = 7$ y $N(S + 1) = 0$, para luego enviarla.

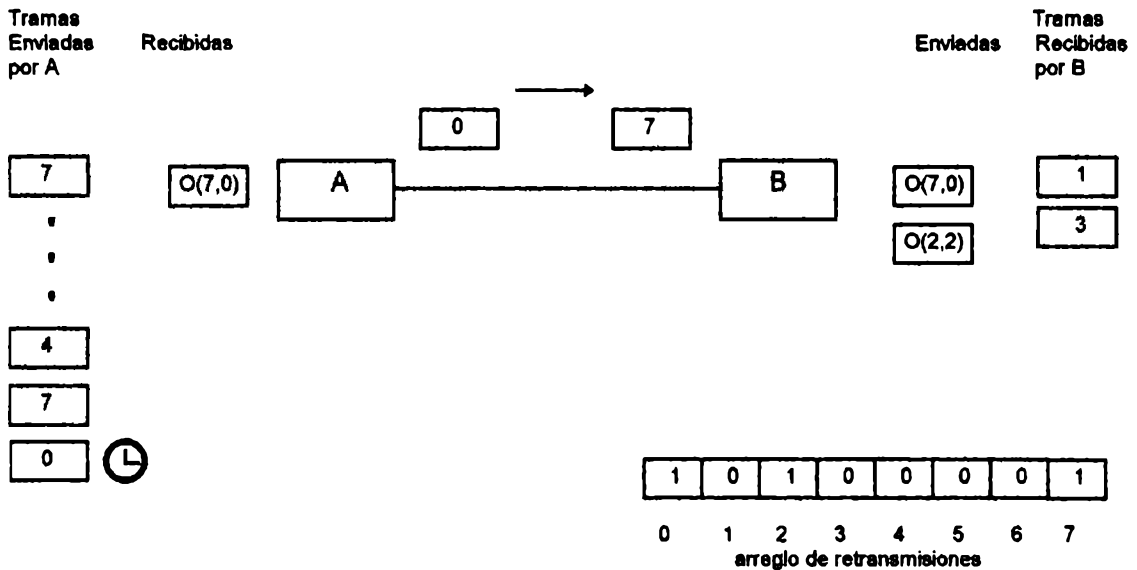


13. En ese instante se reconoce una nueva trama, la 3. Se la guarda en el buffer. En consecuencia, se vuelve a señalar en el arreglo el pedido de la trama 2 con el estado en 1.

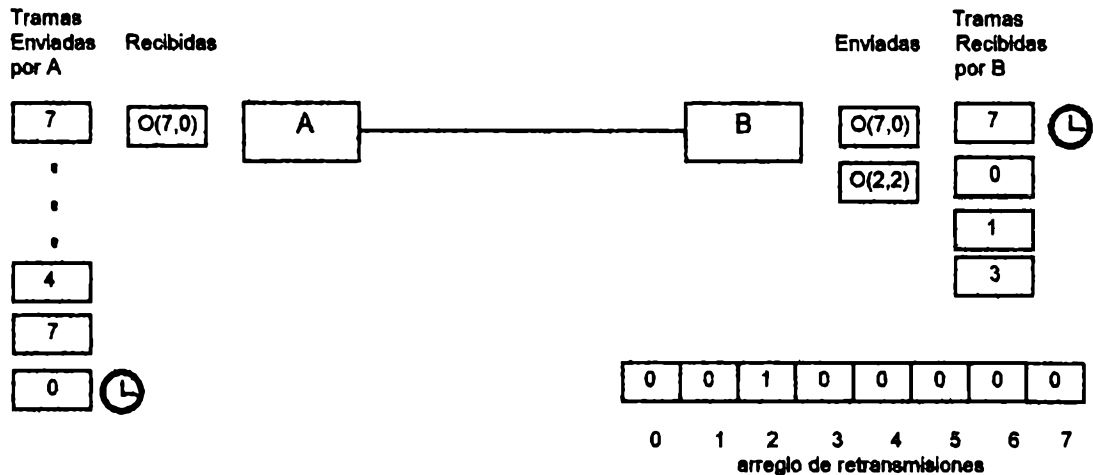
14. Esta nueva trama implica un nuevo pedido de retransmisión con $N(S) = N(S + 1) = 2$.



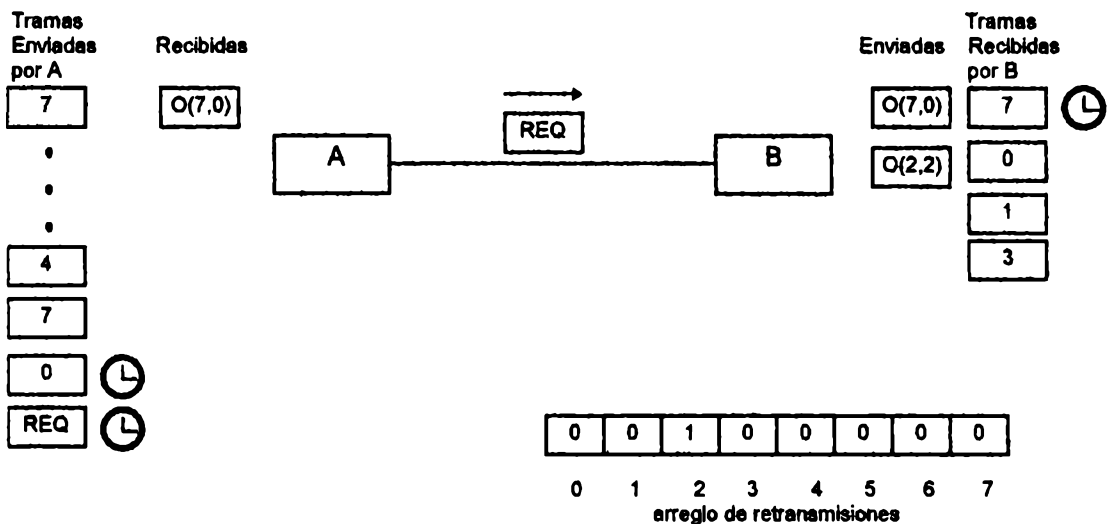
15. Cuando el primer pedido, es decir el correspondiente a las tramas 7 y 0 arriba al emisor, se retransmiten sendas tramas, con la consiguiente reinicialización del temporizador.



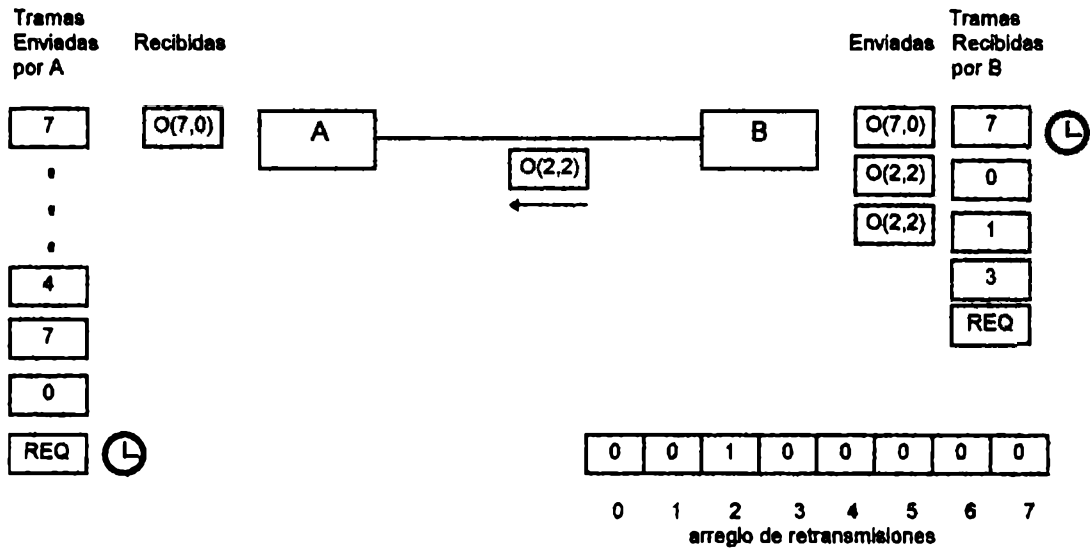
16. Estas tramas le llegan al receptor que almacena sus paquetes en su buffer. Además el estado de ambas tramas pasa a ser 0. Cuando el receptor efectuó un pedido y éste todavía no le fue respondido, no estará en condiciones de transmitirle un RR ya que podría llegar a superponerse con el pedido efectuado anteriormente, no así una trama de información con su correspondiente confirmación.



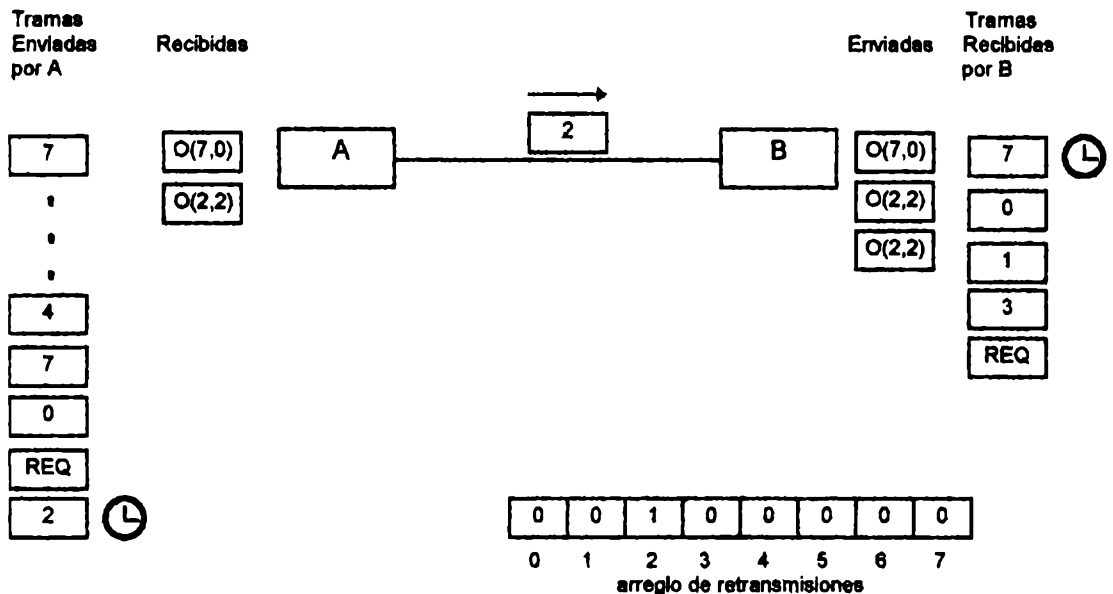
17. El ruido existente en el medio dañó la segunda trama del pedido de retransmisión, lo que le imposibilitó al emisor reconocerla.
18. Al emisor se le vence el temporizador y comienza nuevamente a enviar REQ, VREQ se pone en falso.



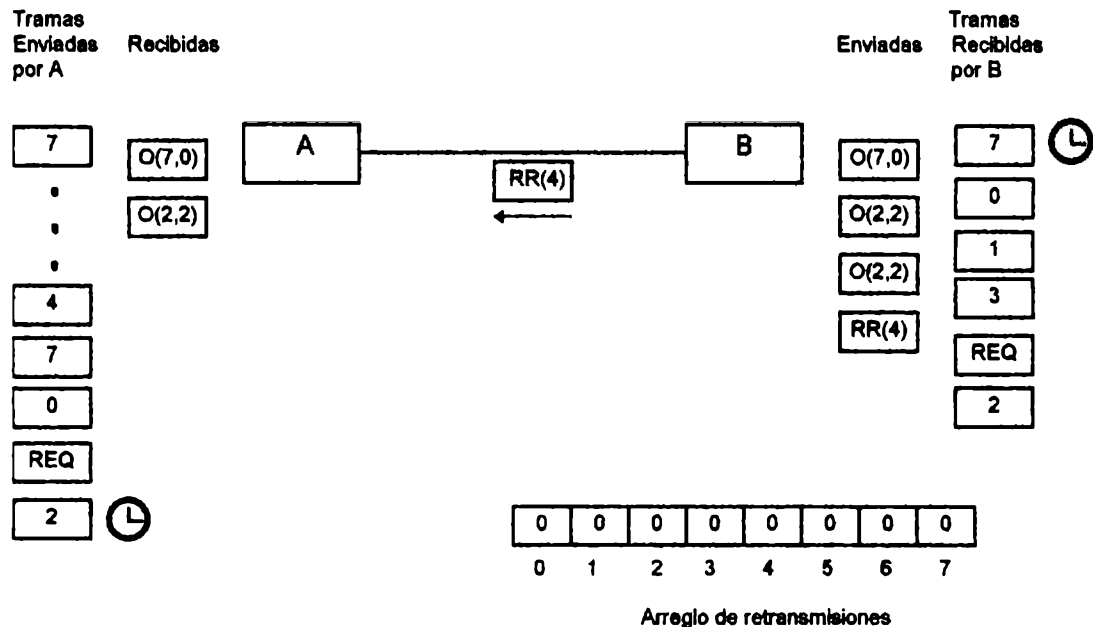
19. El receptor al recibirla, contesta inmediatamente, pero esta vez con el pedido de retransmisión de la trama 2, nuevamente. Si se hubiese enviado un RR(2) con anterioridad, ésta podría haber llegado al emisor como una respuesta al REQ, señalando con ello que se deberían retransmitir todas las tramas a partir de la 2, hecho que no es real. Por lo tanto, como se mencionó anteriormente se prohíbe la transmisión de tramas RR, habiendo pendiente un pedido de retransmisión.



20.El emisor contesta el pedido retransmitiendo la trama 2 poniendo otra vez en marcha el temporizador, coloca VREQ en verdadero.



21.Luego de la llegada de esta trama al nodo receptor, éste estará en condiciones de enviar un asentimiento con $N(R) = 4$. Del mismo modo se resetea el arreglo de retransmisiones.



22. Esta confirmación le permite a la estación emisora correr su ventana y así podrá seguir transmitiendo nuevas tramas. En el caso en que la Capa de Red no se las enviase vencerá el timer del emisor al cual éste deberá obedecer enviando nuevamente REQ con VREQ en falso.
23. Una vez que el receptor se hizo eco del requerimiento enviando nuevamente un RR(4), el emisor deberá retransmitir la trama 4.
24. El receptor almacenará en su buffer el paquete de la trama que se le envió reiniciándolo su timer.

2.3.3 Desconexión del Enlace

El enlace es desconectado luego de la finalización de la transferencia de información o en cualquier momento durante la misma si ocurriese un error fatal, requiriendo del procedimiento. Son diferentes las causas por las que se puede enviar un FRMR, ejemplo de ello son: cuando se reconoce una trama sin errores de CRC pero que no responda a ningún tipo de las existentes, cuando el campo de datos supera la longitud máxima. Ante un comando DISC solicitado por uno de los nodos, el otro deberá contestar con UA, de este modo se termina una desconexión.

2.4 Estructura de la Trama

2.4.1 Propuesta I

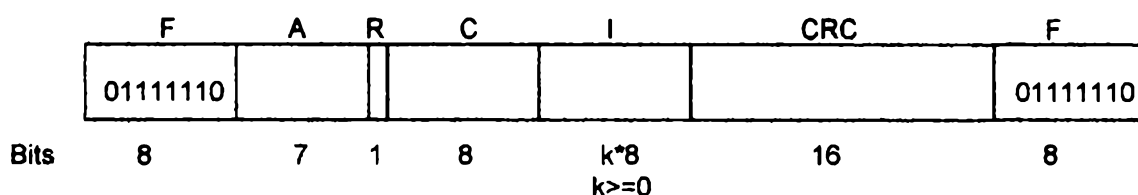


Fig. 16.- Formato de trama

A semejanza de otros protocolos standards cada trama de transmisión tiene el mismo formato específico. Cada trama, como la ilustrada en la fig. 16 está compuesta de:

- Un flag (F) que marca el inicio de la trama.
- Un campo de address o dirección (A) que identifica si se trata de un comando o una respuesta.
- Un bit de pedido de retransmisión (R) que le permite al receptor realizar la solicitud de tramas correspondiente, juntamente con su campo de control.
- Un campo de control (C) que especifica el tipo de la trama con una función específica.
- Un campo opcional de Información que contiene los datos propiamente dichos.
- Un campo de CRC que le permite a la estación receptora efectuar el chequeo de la trama transmitida.
- Un flag (F) que señala el fin de trama.

Flags (F)

Los flags de comienzo y de fin encierran a una trama. Ambas tienen el formato binario 01111110. Siempre que el emisor encuentre cinco bits 1s consecutivos dentro de la trama realiza la inserción de un bit 0 para evitar una confusión con los flags. Del mismo modo el receptor luego de detectar la trama comprendida entre los dos patrones de bits especiales o flags, y al notar cinco bits con valor 1 seguidos de un 0 deberá substraerlo. El flag de finalización puede ser utilizado como flag de comienzo para la próxima trama.

Campo de Address (A)

Este campo de 7 bits señala si se trata de un comando o de una respuesta en direcciones opuestas. Su acción es redundante en la distinción ya que va acompañada por el tipo de trama en el campo de control, no obstante actúa como un chequeador de error adicional permitiendo la diferenciación del flujo de información que viaja en sentidos opuestos y en consecuencia puede detectar un loop back accidental.

Suponiendo que tenemos las estaciones A y B :

- 1100000 = 3 : Comandos de B a A y Respuestas de A a B.
- 1000000 = 1 : Comandos de A a B y Respuestas de B a A.

Este campo al igual que el campo de control se transmiten en orden inverso.

Pedido de Retransmisión (R)

Este bit que si bien podría ser considerado como el octavo bit del campo de Address, le permite a la estación receptora efectuar la solicitud de tramas que se detectaron erróneamente o que no pudieron reconocerse, cuando se halla en 1. Se la utiliza en conjunción con el campo de control que define la función del pedido. En los otros casos su valor se mantiene en 0.

Campo de Control (C)

Indica el tipo de trama y por lo tanto la función que deben cumplir. Se tienen cuatro formatos: No numeradas (U), Supervisoras (S), Información (I) y de Pedido de retransmisión (O), las mismas se muestran en la fig. 17.

1. **No numeradas (U)** : Se utilizan para el establecimiento y desconexión del enlace de datos. Estas tramas no son chequeadas secuencialmente por lo que no cuentan con el bit SEQ ni con los contadores N(S) ni N(R). Se indica con los dos primeros bits con 1. Seis bits del campo de control permiten codificar sus comandos y respuestas. Se constan de distintos tipos como ser:
 - 1.1. **111100 = SABM** : Se usa para la inicialización y reinicialización del enlace.
 - 1.2. **001110 = UA** : Se envía para confirmar la aceptación de un comando no numerado.
 - 1.3. **111000 = DM** : Se la utiliza para indicar que el receptor se desconecta lógicamente del enlace.
 - 1.4. **001010 = DISC** : Se hace uso de ella cuando un extremo quiere desconectar el enlace. Una vez que el otro punto haya recibido el comando lo deberá confirmar con UA.
 - 1.5. **101001 = FRMR** : Cuando se recibió una trama que no puede ser reconocida pese a que no se detectaron errores de CRC, el receptor está desorientado con respecto al estado en que se encuentra el otro extremo. Por lo tanto envía esta trama, originando el reseteo de todas las variables de estado, como el arreglo de retransmisiones. Posteriormente deberá enviarse DISC o SABM, para terminar o reiniciar una nueva conexión.

2. **Supervisoras (S)** : Asisten en la transferencia de datos, aunque no llevan datos puros en sí. El campo comienza con los bits 1 y 0 seguido de un código de 2 bits que señala su propósito:
 - 2.1. **00 = RR** : Se la utiliza para distintos fines, siendo la principal consistente en confirmar tramas de información si no hubo un asentimiento superpuesto por piggybacking. Para ello consta de un campo $N(R)$ el cual indica la próxima trama que espera recibir confirmando en consecuencia hasta la trama $N(R) - 1$. Además se halla presente el bit SEQ que señala a que grupo de secuencia pertenece la trama esperada. También se hace uso de un RR cuando no se tienen tramas de información para enviar pero se quiere mantener el enlace activo. Otra tarea es la de volver al estado de ready cuando se envía previamente un RNR.
 - 2.2. **10 = RNR** : Es usada por el receptor para indicarle al emisor que se encuentra temporalmente ocupado, posiblemente por un problema de escasez de memoria y le solicita que deje de enviarle tramas. La transferencia se reanuda con RR, un Pedido de Retransmisión o con alguna otra trama de control. La estructura con respecto a los campos es similar al RR.
 - 2.3. **01 = ACC** : El receptor envía esta trama con la finalidad de aceptar una trama indicada en $N(R)$ y perteneciente a la secuencia determinada por SEQ. Se pone especial énfasis señalando que la transmisión de la misma no ejecuta las mismas funciones que un RR, si no que si bien se acepta la trama no se desecha del buffer el paquete correspondiente a ella y su principal propósito es la de desactivar el temporizador asociado en el emisor.
3. **Información (I)** : Se hacen uso de ellas para indicar que se trata de una trama del mismo nombre. Se la reconoce por tener como primer bit al 0, además contiene los contadores $N(S)$, $N(R)$ y un bit SEQ. El primero se utiliza con el objeto de asegurar que las tramas sean recibidas en el orden apropiado y se usa el $N(R)$ para confirmar tramas de información recibidas por piggybacking. La tarea del bit SEQ es la de evitar la superposición de ventanas, con tramas correspondientes a dos secuencias consecutivas, mediante la alternancia del bit de 0 a 1 y viceversa. Es decir, $N(S)$ indica el número de la trama de información dentro de la secuencia SEQ y $N(R)$ la próxima esperada.
4. **Pedido de Retransmisión (O)** : este campo indica el rango de tramas que el receptor solicita conjuntamente con el bit del pedido de retransmisión (R). Además del bit 0 contiene los campos $N(S)$ y $N(S + 1)$, donde el primero señala la primer trama del rango que solicita y el segundo la última. Si se tratase de una única trama ambos especifican a la misma. El significado del bit SEQ se asocia a $N(S)$ ya que para $N(S + 1)$ puede ser deducido.

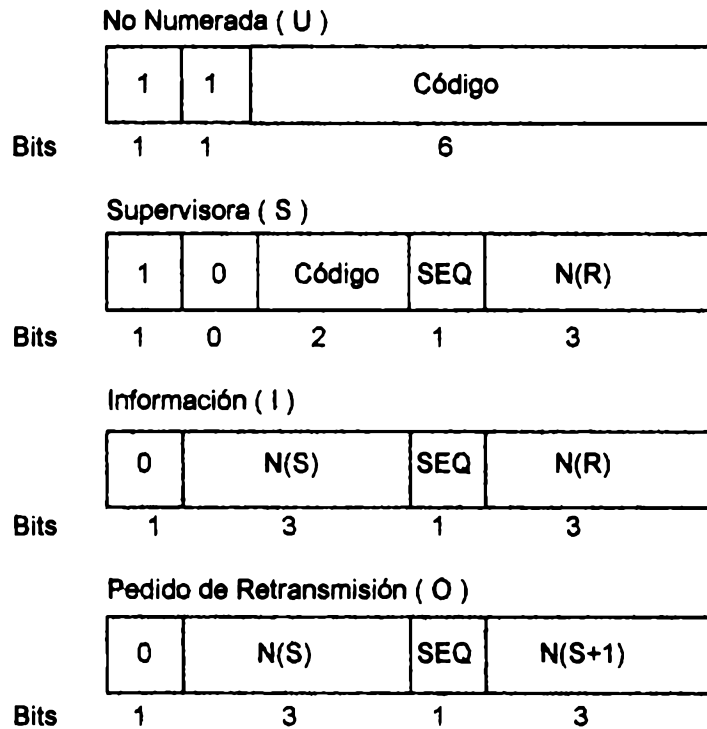
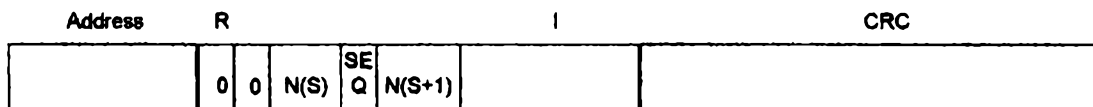


Fig. 17 - Campos de Control

Un pedido de retransmisión se hace efectivo enviando una trama constituida por el bit y el campo en cuestión con los valores 1 y los antes mencionados respectivamente.

Información (I)



Pedido de Retransmisión (O)

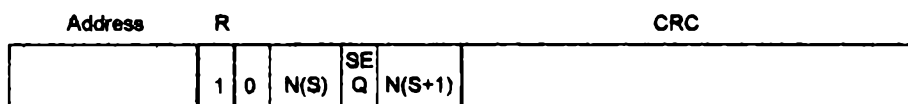


Fig.18 - Distinción entre Tramas de Información y de Retransmisión

Cabe destacar que aunque los campos de control de información (I) y de pedido de retransmisión (O) tienen la misma estructura, la distinción estará dado por el bit (R), que en el primer caso siempre se enviará con 0, mientras con el segundo será 1. Gráficamente se ilustra en la Fig. 18.

Campo de Información

Se utiliza para transportar la información en sí concerniente al enlace de datos que se refleja en el campo de control de igual nombre. De hecho las otras tramas no cuentan con este campo. La longitud puede ser variable si bien debe respetar ser un múltiplo de 8 bits.

Campo de CRC

Es el código de redundancia cíclica de 16 bits. Permite al receptor chequear errores que podrían haber surgido durante la transmisión sobre una trama recibida. El emisor calcula el CRC en base al polinomio generador $x^{16} + x^{12} + x^5 + 1$, es decir el perteneciente a la norma internacional CRC-CCITT y luego lo anexa a la trama. En consecuencia cuando una trama es reconocida en el receptor, es dividida por el mismo polinomio y si la misma no presentase error, se le quitan los 16 bits.

Tabla de Comandos y Respuestas de la Propuesta I

Formato	Comandos	Respuestas
No numeradas	SABM	UA
		DM
	DISC	
		FRMR

Fig.19 .-Comandos y respuestas de la propuesta I

Formato	Comandos	Respuestas
Información	I	
Supervisoras	RR RNR	RR RNR ACC
Pedido de retransmisión	O	

Fig.19 .-Comandos y respuestas de la propuesta I (Continuación)

2.4.2 Propuesta II

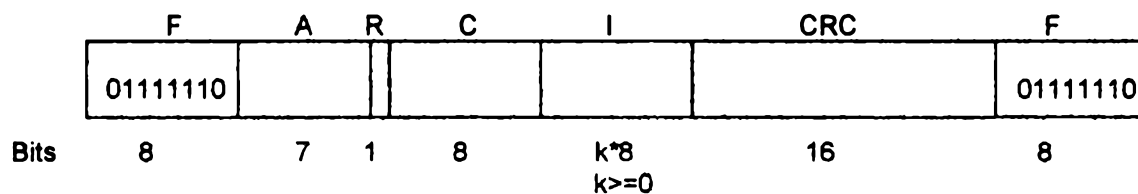


Fig. 20 - Formato de trama

Tiene una estructura de la trama similar a la propuesta anterior, mostrada en la fig.20. La diferencia se da en la ausencia de tramas supervisoras de tipo ACC. En su lugar se cuentan con otras también del mismo grupo denominadas REQ. Además de ello, el campo SEQ se ve modificado mediante el bit P/F, mencionado anteriormente. Los campos de control se ilustran en la fig.21.

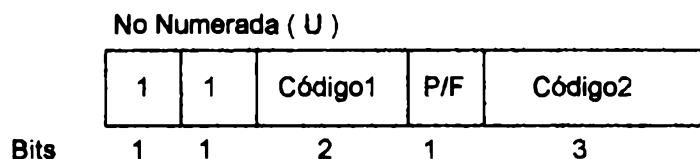


Fig. 21 - Campos de Control

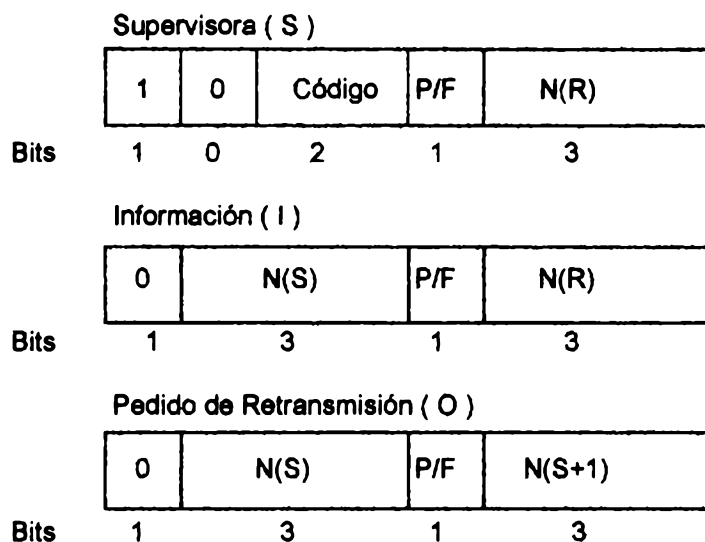


Fig. 21 - Campos de Control (Continuación)

- **01 = REQ** : Sirven para cuando el emisor terminó de enviar todas las tramas pertenecientes a su ventana o no tiene más tramas para enviar y pasado cierto lapso de tiempo el receptor no muestra indicios de actividad. La finalidad del envío de esta trama consiste en conocer la situación de las tramas que se enviaron ante una pérdida o daño de la contestación de parte del receptor. No lleva N(R) ni SEQ, es decir los bits correspondientes quedan postergados para un futuro uso.

Tabla de Comandos y Respuestas

Formato	Comandos	Respuestas
Información	I	
Supervisoras	RR RNR REQ	RR RNR
Pedido de retransmisión	O	

Fig.22 - Comandos y Respuestas de la Propuesta II

Formato	Comandos	Respuestas
No numeradas	SABM	UA
		DM
	DISC	
		FRMR

Fig.22 - Comandos y Respuestas de la Propuesta II (Continuación)

2.5 Análisis Comparativo de las Propuestas

Dada una descripción detallada de cada una de las propuestas, se impone un análisis comparativo de las mismas, destacando tramas empleadas y controles que se realizan examinando la performance de ambas.

El desarrollo de las propuestas se fundamentó en mejorar la retransmisión selectiva, siendo una de las consecuencias la ampliación de la ventana de recepción, en comparación a los protocolos que se utilizan en la actualidad.

En las dos propuestas planteadas se mantuvo dicha idea, sin embargo se reconocen diferencias notables en la gestión. En la primer propuesta se conservó su perfil tradicional, mientras que en la segunda el diseño es menos usual.

Seguidamente se resumen las características más salientes de las dos ideas.

Propuesta I

- timer por trama en emisor
- usa bit SEQ
- usa trama ACC
- no usa trama REQ
- retransmisiones de tramas de información a tiempos constantes controlados por cada timer

Propuesta II

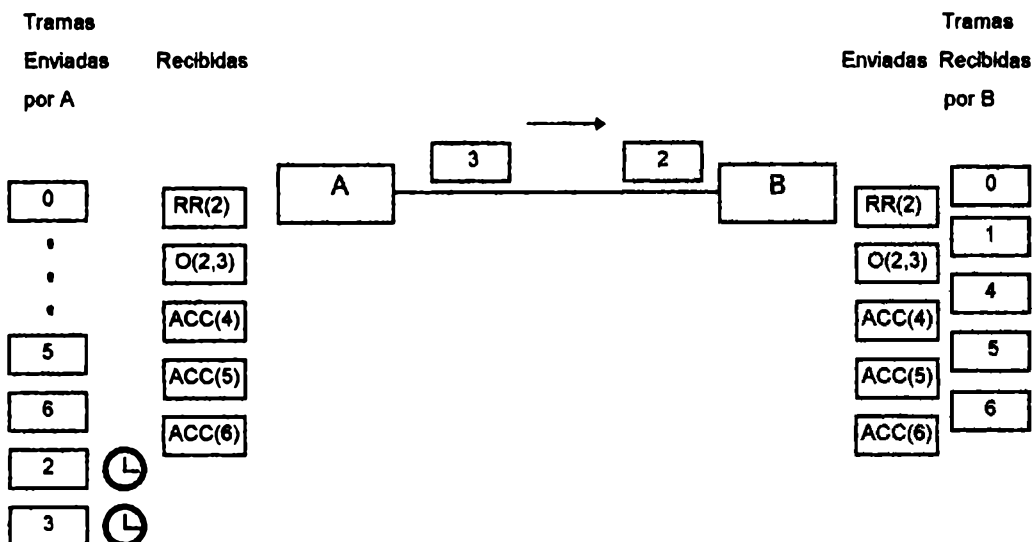
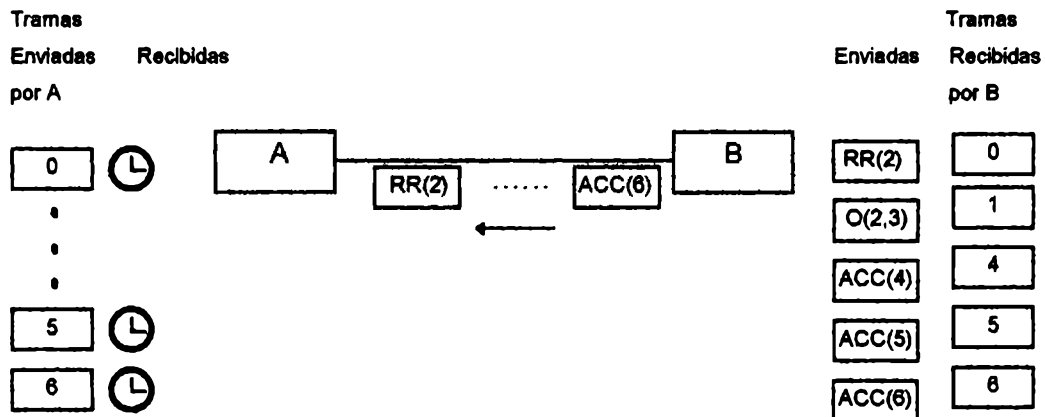
- un timer en emisor
- no usa bit SEQ
- no usa trama ACC
- usa trama REQ
- envío de tramas REQ a tiempos constantes

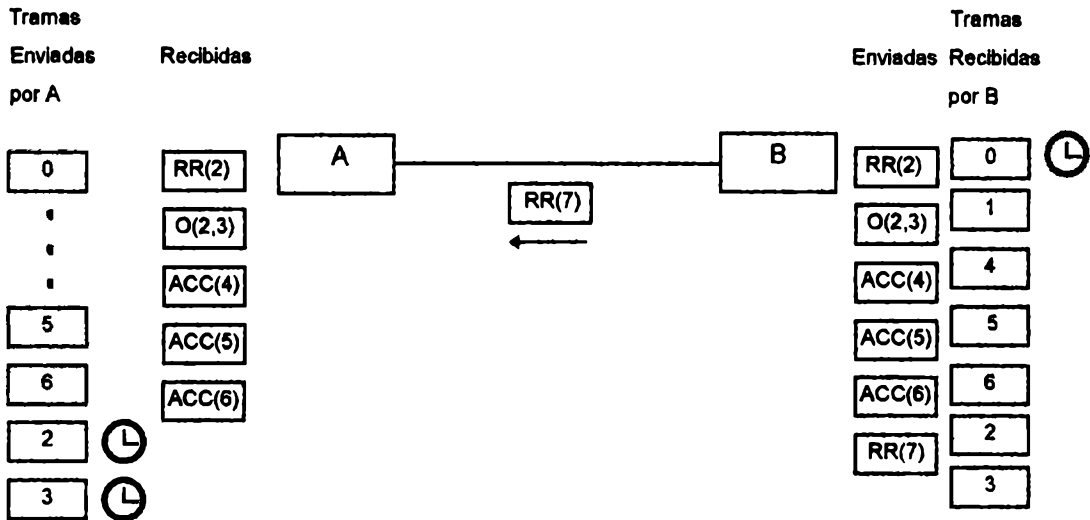
La propuesta I usa tantos timers como tramas enviadas pendientes de confirmación, ésto se traduce en mayor cantidad de memoria y procesamiento, mientras tanto la propuesta II utiliza solamente un único temporizador efectuando un control similar.

El hecho de usar el bit de SEQ implica, si bien un mejor control de las tramas que se envían un procesamiento extra en el receptor.

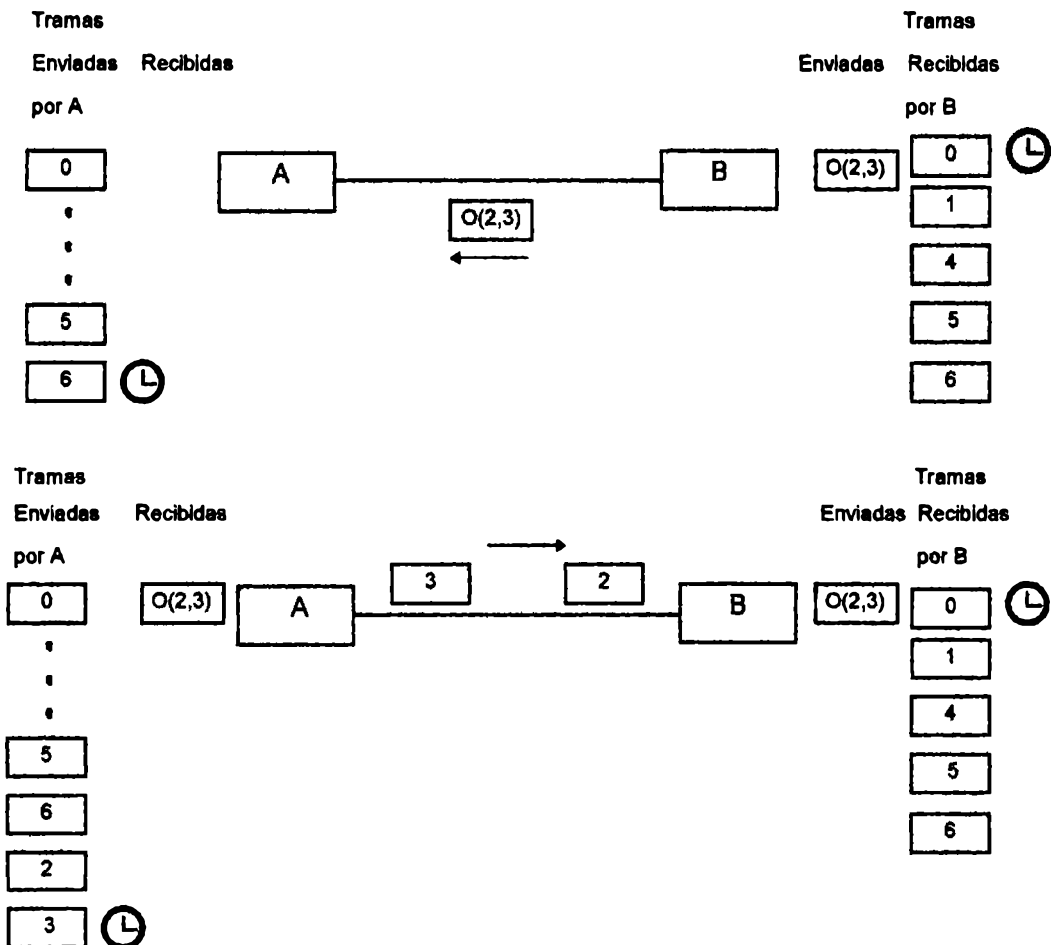
A continuación se verán los distintos casos de retransmisión apoyados en sendos ejemplos. Suponiendo que se envían las tramas 0, 1, 2, 3, 4, 5 y 6, y le llegan al receptor la 0, 1, 4, 5, y 6, ¿cuáles son las acciones que le demandarían a una y otra propuesta?

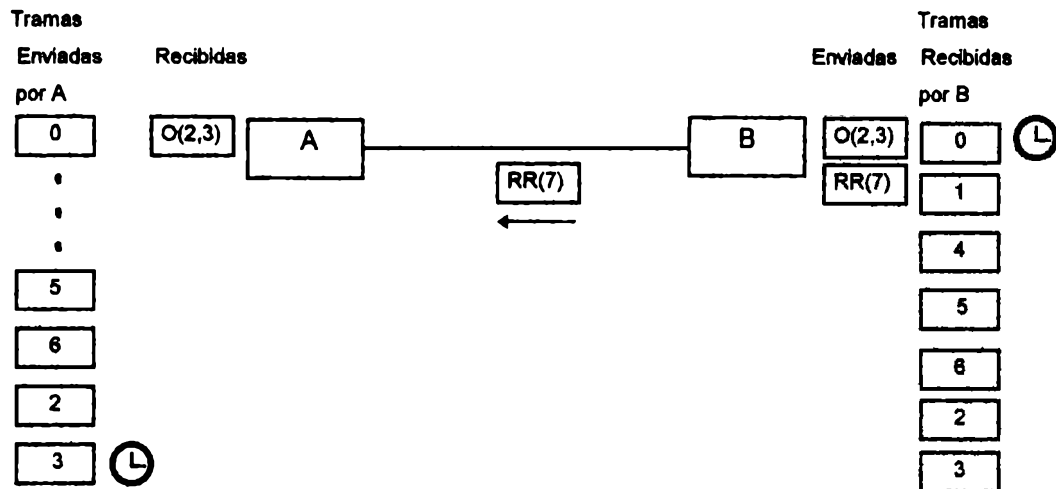
- En la propuesta I, el receptor confirmará las tramas 0 y 1 con una trama de información con $N(R) = 2$ o un $RR(2)$ en su defecto. Enviará un pedido de retransmisión para las tramas 2 y 3, $ACC(4)$, $ACC(5)$ y $ACC(6)$. Posteriormente a la llegada de las tramas 2 y 3 retransmitidas confirmará las tramas pendientes. En resumen, el receptor transmite una cantidad de seis tramas.





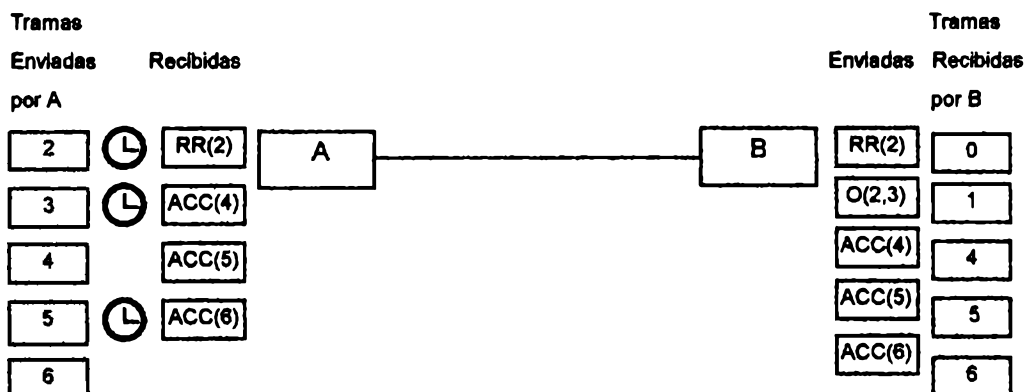
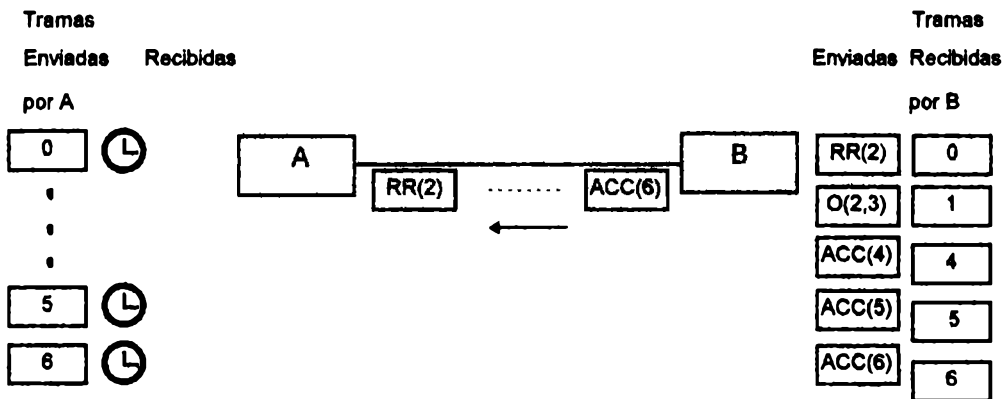
- En la propuesta II, el receptor transmitiría un pedido de retransmisión para las tramas 2 y 3. Luego del arribo de dichas tramas, confirmaría todas las postergadas. Para la misma situación, esta propuesta utiliza solamente dos tramas, con lo cual se evidencia que tiene una mejor performance en la respuesta a este ejemplo.



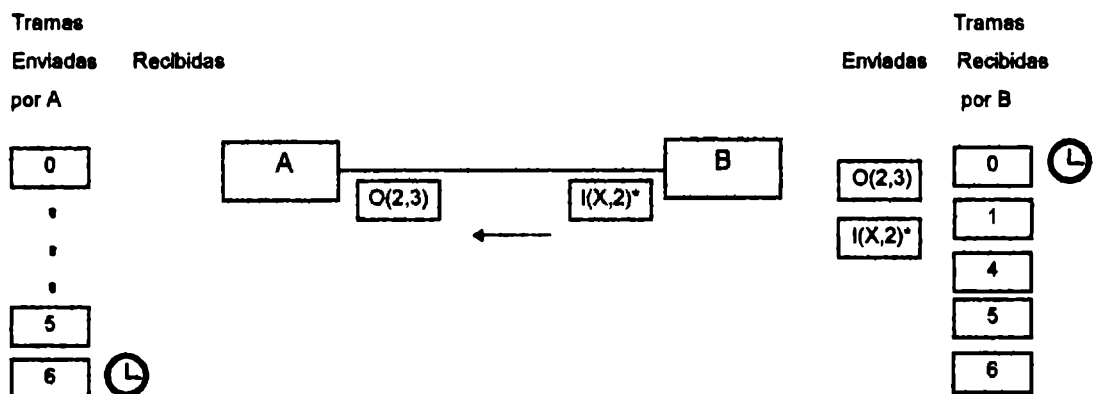


Presumiendo que podrían llegar a perderse o dañarse alguna/s de la/s trama/s que solucionan este hecho podemos analizar que ocurriría en los distintos casos. En la propuesta I, el no reconocimiento de alguna de las tramas ACC o un RR traería como consecuencia el disparo de esa trama por timeout, si bien no trae efectos perjudiciales tampoco es deseable que ocurra.

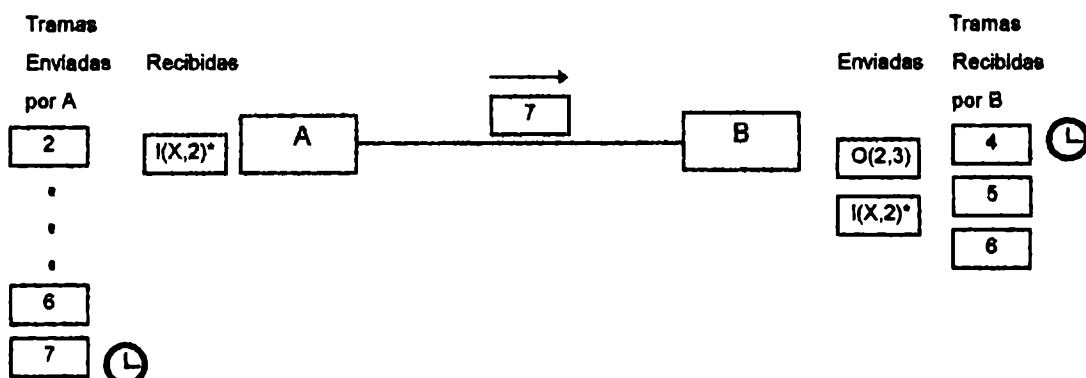
- Si en cambio la que se perdiese fuese el pedido de retransmisión, aunque las otras tramas de control hayan llegado al emisor, éste no retransmitirá tramas.

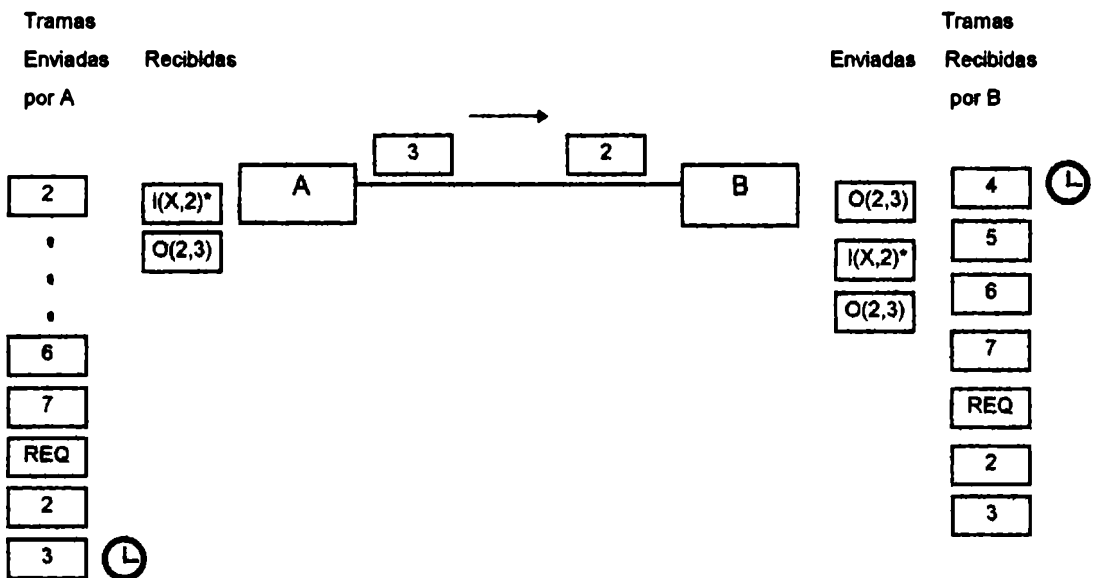
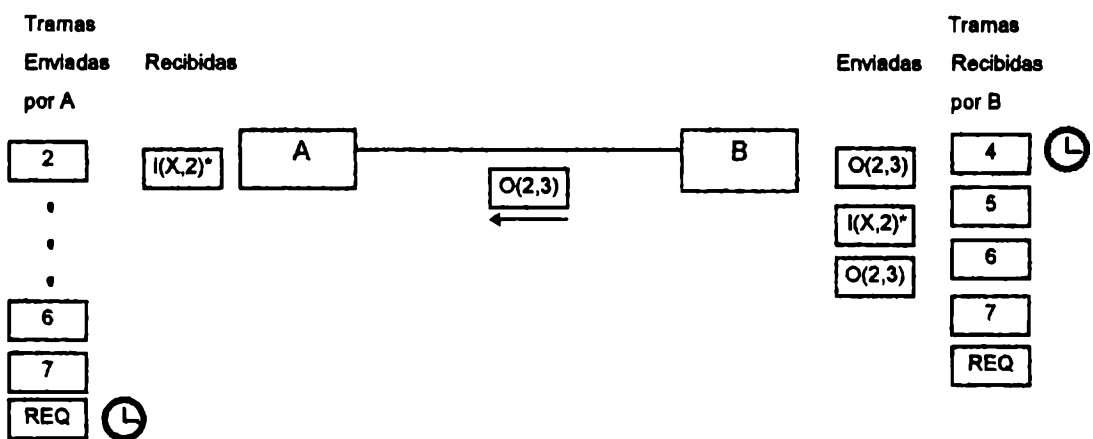
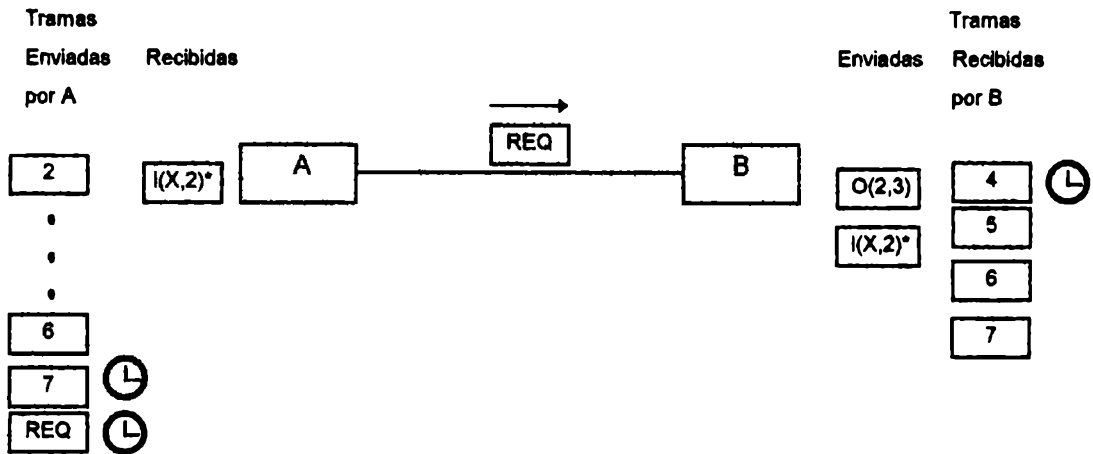


- En la propuesta II, la no identificación de la misma trama tendría otras derivaciones. Suponiendo que el receptor tiene sus propias tramas de información para enviarle al emisor, éste le superpondrá el asentimiento de las tramas 0 y 1 con $N(R) = 2$. El emisor la recibe, modificando su ventana, pero no sabe hasta el momento que le ocurrieron a las tramas subsiguientes a la 2. Sin embargo, puede proseguir con la entrega de nuevas tramas al receptor, hasta acabar con la ventana o parar la emisión si la capa superior no le manda más paquetes. Luego de pasado un tiempo, el emisor le solicita al receptor indicarle cual es su estado transmitiendo un REQ. Antes de recibir dicha trama, el receptor envía una nueva trama de información, con $N(R) = 2$. Inexorablemente esta trama llega al emisor antes que la verdadera contestación al REQ. Aquí surge el problema. El hecho de tomar como respuesta del REQ una trama de información con $N(R) = 2$ no significa lo mismo que un pedido de retransmisión de las tramas 2 y 3. En el primer caso, el emisor retransmitiría todas las tramas de su ventana a partir de la 2 en adelante. Esto dejaría de ser una retransmisión selectiva. Para solucionar este inconveniente, no se tomaría como una respuesta válida para un REQ una trama de información. Tampoco se dice que se trata de una conclusión muy satisfactoria, ya que si en realidad lo que se pretendía era contestar con un RR con los beneficios del piggybacking se hubiese ganado en tiempo y espacio.



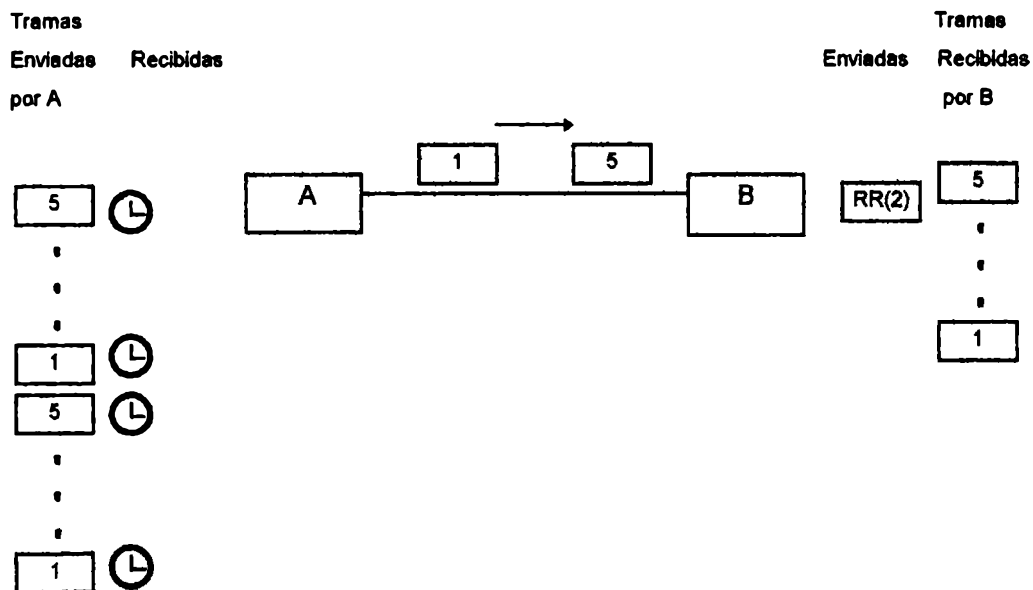
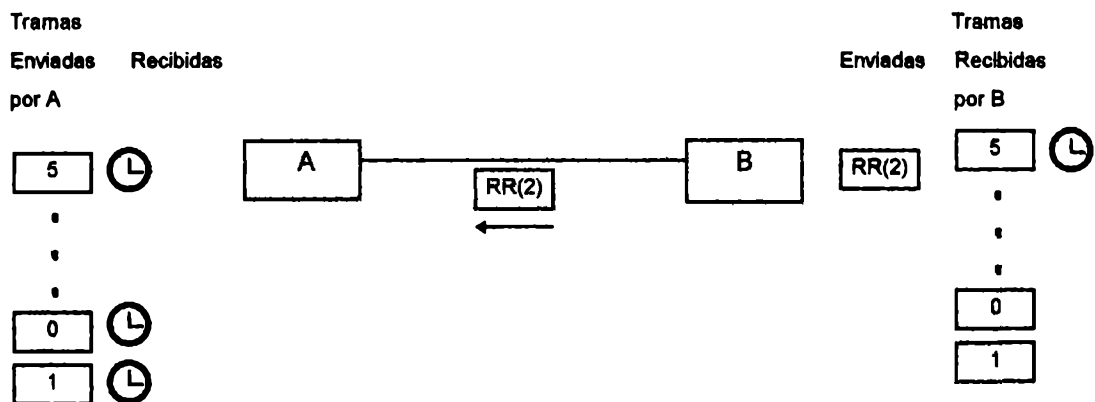
* Trama de información siendo X el número de secuencia de la trama enviado





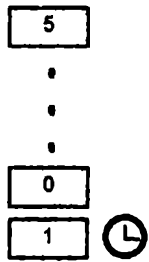
Observando otro caso en donde el emisor transmite las tramas 5, 6, 7, 0 y 1, el receptor las recibe a todas confirmándolas con RR(2), suponiendo que por el momento no hay tramas de información de receptor a emisor. El asentimiento se pierde.

- Teniendo en cuenta la propuesta 1, el emisor comenzará a retransmitir por vencimiento de temporizador. Si los tiempos de puesta en línea fueron muy cercanos, los timeouts se darán uno detrás de otro desperdiciando de este modo muchas retransmisiones ya que cuando el receptor se disponga nuevamente a enviar un RR(2) y éste le llegue al emisor, ya habrán llegado a destino una cierta cantidad de tramas retransmitidas.

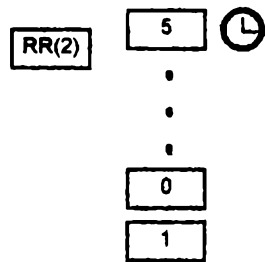


- Con la propuesta II, esto no ocurriría ya que se enviaría un REQ si se vence el timer de emisión, a un tiempo coherente estadísticamente determinado que no agobie al receptor. De igual modo, aunque los temporizadores se venzan a tiempos muy distantes, siempre la construcción de una trama llevará mayor procesamiento que una trama standard REQ.

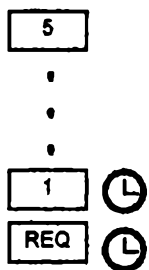
Tramas
Enviadas Recibidas
por A



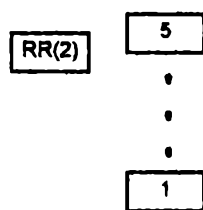
Tramas
Enviadas Recibidas
por B



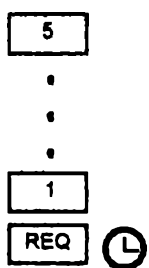
Tramas
Enviadas Recibidas
por A



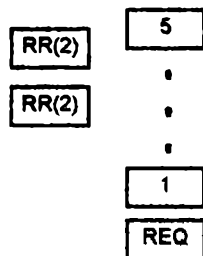
Tramas
Enviadas Recibidas
por B



Tramas
Enviadas Recibidas
por A

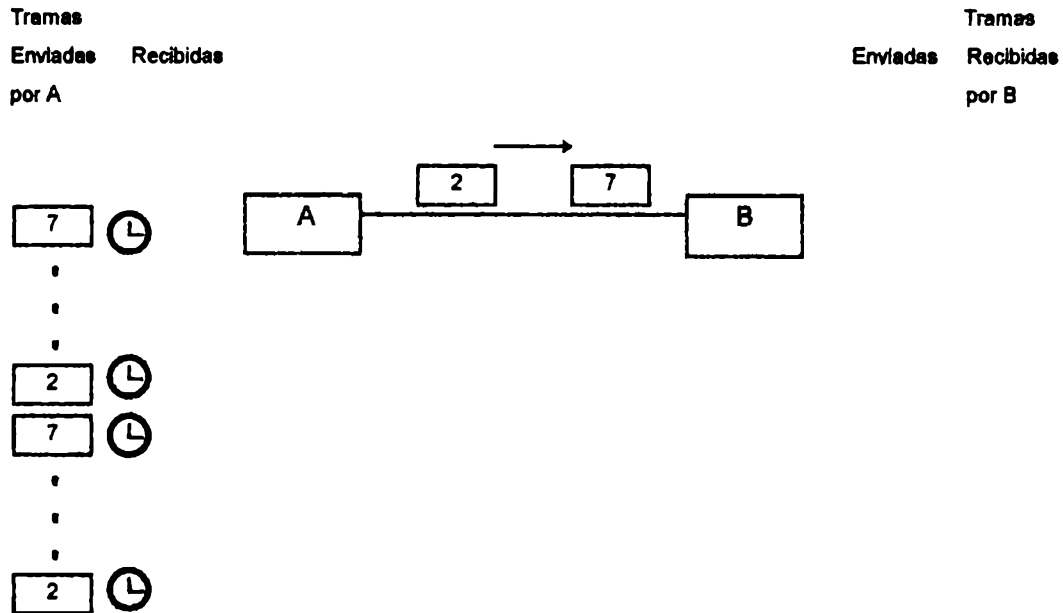


Tramas
Enviadas Recibidas
por B

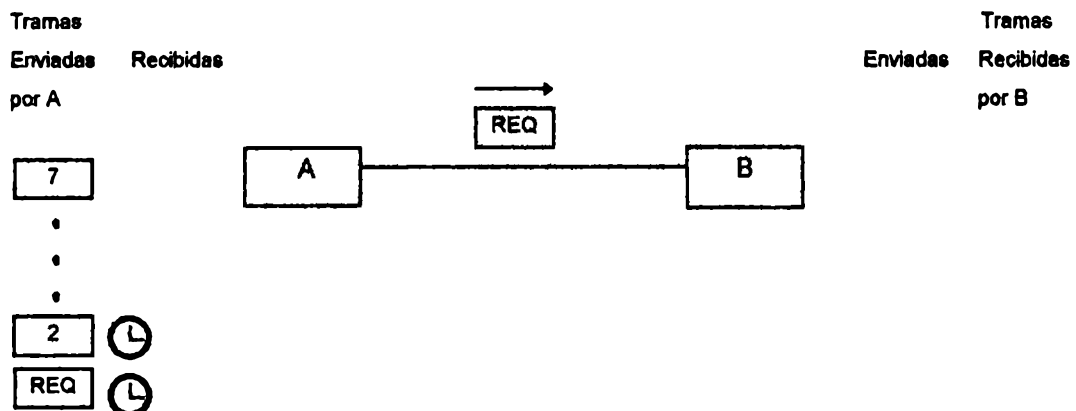


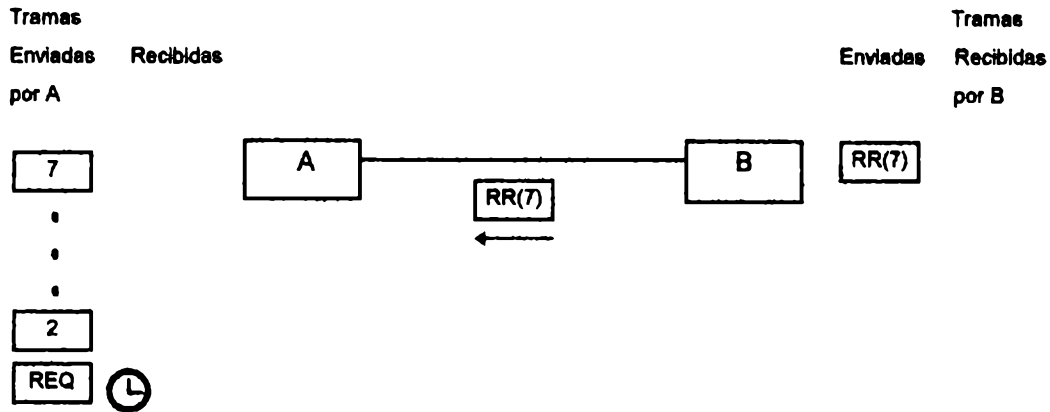
Consecutivamente se plantea otra cuestión en donde el emisor envía las tramas 7, 0, 1 y 2. Sin embargo el ruido existente en el medio hace que se pierdan todas las tramas.

- La propuesta I, retransmite todas a partir de los timeouts.



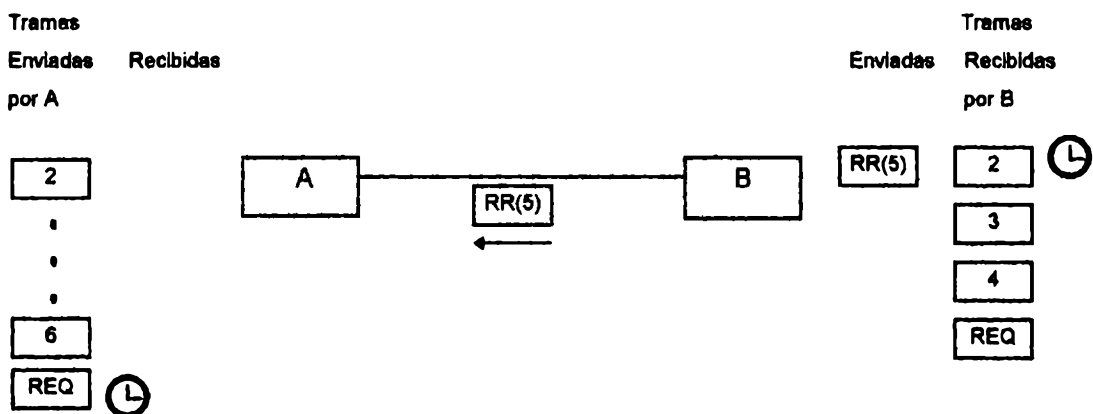
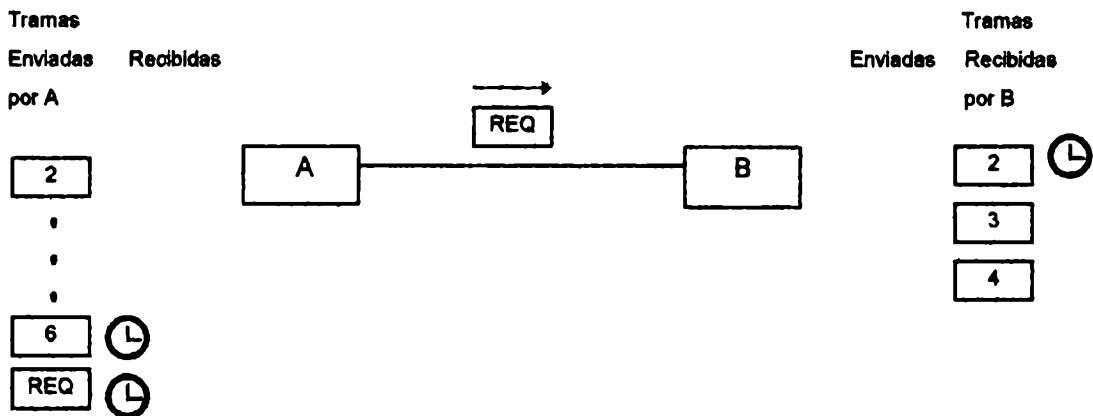
- La propuesta II, vuelve a emitir no a ciegas sino que primero investiga que ocurrió en la otra punta, es decir luego del vencimiento del temporizador reseteado en última instancia con la emisión de la trama 2 dispara un REQ y en base a ello retransmite. Tal vez implique en una pérdida de tiempo por el envío de REQ y RR, con respecto a la anterior, pero también puede traducirse en un control más estricto como se puede ver claramente a continuación.

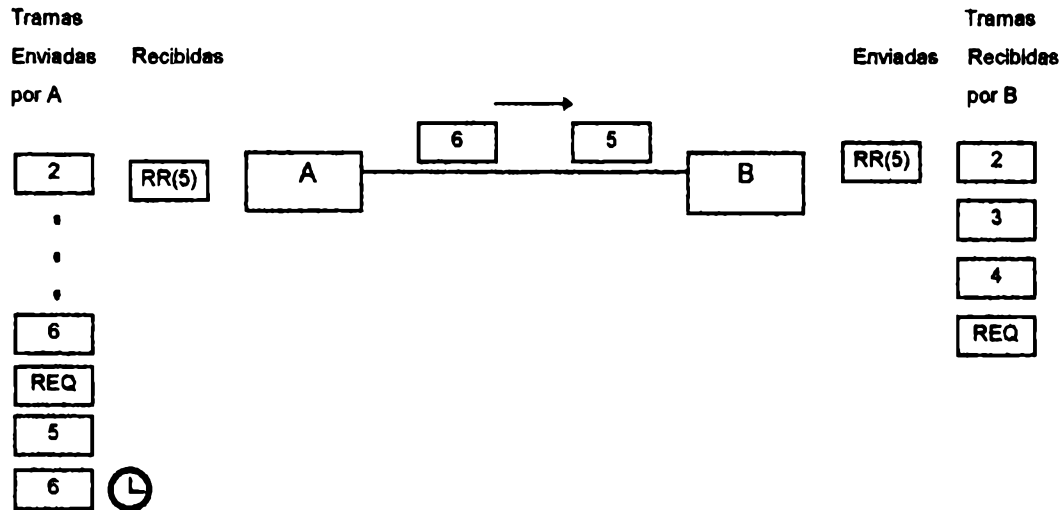




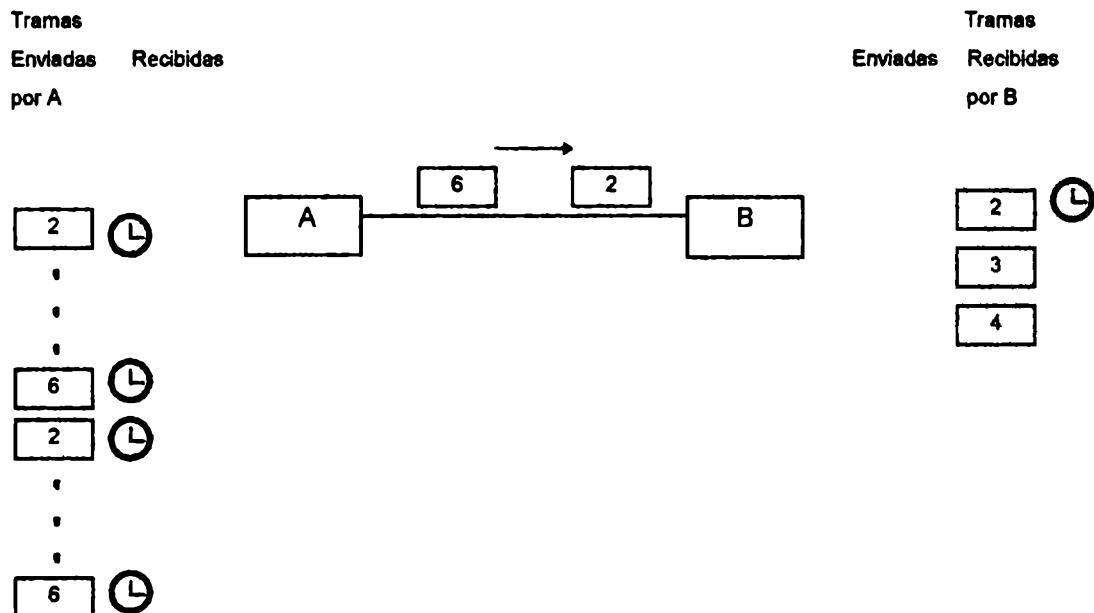
Enviadas las tramas 2, 3, 4, 5 y 6, ahora se reciben las tres primeras.

- Como se dijo precedentemente, en la última propuesta el emisor indaga primero al receptor y ante su respuesta toma su decisión en una situación límite. Es decir, luego del REQ el receptor contesta con RR(5), lo que le indica al emisor que le debe transferir las tramas 5 y 6 nuevamente.





- En el caso de la propuesta I, como hay una retransmisión por timeout, en el peor de los casos como se explicó con anterioridad, podría llegarse a una consecuencia no muy deseada si se retransmitirían todas las tramas de la ventana actual. De modo que se volverían a emitirse tres tramas de la 2 a la 4 sin que hayan sido necesarias enviarlas.



2.6 Conclusión

De la comparación realizada se desprende que si bien ambas propuestas alcanzan el objetivo por el cual fueron diseñadas, difieren substancialmente en su gestión.

Si bien una transferencia de datos exitosa resulta similar, no lo es si se tiene en cuenta la cantidad de temporizadores que deben mantener una y otra propuesta.

Ante una situación patológica, el número de tramas que usa la propuesta I, es superior a la que utiliza la propuesta II. Un mero ejemplo de ello es un pedido de retransmisión.

La pérdida de una confirmación, posterior a un envío de tramas provoca en la propuesta I la retransmisión de todas aquellas a las que se les venció el timer asociado, mientras que en la propuesta II con la transmisión del REQ genera por un lado que el emisor permanezca en espera de la respuesta del receptor, pero también que se reemitan solamente las que el destinatario requiera.

La ventaja primordial de la propuesta II sobre la I, es que realiza una investigación del problema que le está sucediendo al receptor y en relación a ello ejecuta la retransmisión. En la propuesta I, ésto no ocurre ya que el emisor envía nuevamente por expiración de los temporizadores.

Por ello se considera más eficiente la propuesta II. Los puntos más relevantes de la misma serán modelizados usando RdP.

Capítulo III

ESPECIFICACIÓN CON REDES DE PETRI

3. Especificación del Protocolo

Los protocolos reales y los programas que los desarrollan son, por lo general, bastante complicados, por lo cual se han realizado distintas investigaciones con el objeto de encontrar técnicas matemáticas para especificar y verificar protocolos. Se cuentan con diferentes técnicas y modelos, ejemplo de ellos son las máquinas de estados finitas y las redes de Petri. También es posible utilizar lenguajes de especificación tales como Estelle o Lotos.

Una red de Petri (RdP) es una herramienta matemática que puede servir para modelar el comportamiento de sistemas de naturaleza muy diferente, en especial se describirá esta técnica con el fin de especificar los puntos principales del protocolo II.

3.1 Generalidades

Una red de Petri es un grafo orientado en el que intervienen dos clases de elementos: lugares y transiciones representados por circunferencias y segmentos rectilíneos respectivamente, unidos alternativamente por arcos. Un lugar puede tener un número positivo o nulo de marcas o tokens. La capacidad determina el número máximo de tokens que se puede mantener en un lugar. Una marca se representa por un punto en el interior de un lugar. El conjunto de marcas en un instante dado a cada uno de los lugares se denomina marcado de una red de Petri.

Para la descripción de sistemas concurrentes, a los lugares se les asocian acciones o salidas del sistema y a las transiciones eventos y acciones o salidas.

La evolución del marcado de una RdP consiste en pasar tokens de un lugar a otro a través de distintas transiciones.

- un lugar p es un lugar de entrada a una transición t si existe un arco orientado de p hacia t ;
- un lugar p es un lugar de salida de una transición t si existe un arco orientado de t hacia p ;
- una transición está sensibilizada si todos los lugares de entrada a la misma están marcados.

Una transición sensibilizada es disparada si el estado asociado a ella se verifica. El disparo consiste en quitar un token a cada uno de los lugares de entrada y en agregar uno a los lugares de salida.

El estado asociado a una transición indica una cierta condición para el disparo de la misma. Si la RdP es temporizada, cada transición tiene asociado un valor entero que indica un cierto tiempo de retardo para que se dispare la transición desde que ésta es sensibilizada. Las transiciones pueden tener también condiciones de tipo booleano. Es decir, una transición puede tener asociada una comparación de algún tipo.

Las transiciones pueden tener arcos inhibidores que les llegan desde los nodos. Estos arcos indican la operación de negación. Esto es, un arco inhibidor sensibiliza una transición si no tiene ninguna marca, si tuviera alguna no la realiza.

Los arcos examinadores permiten efectuar un testeo disparando una transición pero sin la transferencia de tokens.

3.2 Modelización del Protocolo

Para la modelización se tomaron los aspectos más relevantes y novedosos del protocolo. En consecuencia, se obviaron situaciones que de hecho son importantes como el envío de RNR, FRMR o el análisis de una trama para detectar errores de CRC. Sin embargo como estas circunstancias se contemplan y son de uso en los protocolos standards, no se modeló al respecto.

A continuación se presentarán los lugares y transiciones asociados a las situaciones, aclarando los distintos estados que puede ir tomando la RdP. Para la distinción de arcos inhibidores y examinadores, se utilizaron arcos de colores diferentes, negros para los primeros y azules para los últimos.

3.2.1 Envío de Paquetes

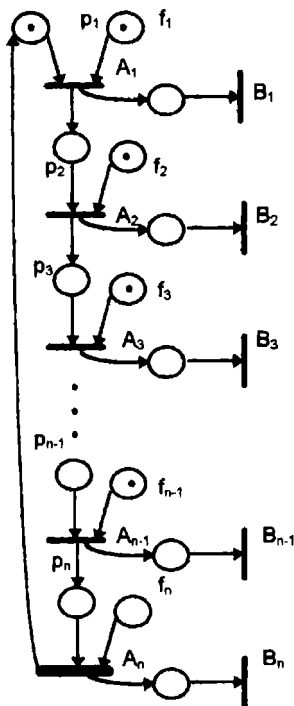


Fig. 23 - Envío de Paquetes

La Capa de Red le envía paquetes a la Capa de Enlace del emisor. Este debe armar las tramas de información con los datos necesarios para luego transmitirlos.

Como se muestra en la fig. 23 las transiciones A_1, \dots, A_n representan la recepción de paquetes de la Capa de Red; f_1, \dots, f_n son lugares que indican a los paquetes y p_1, \dots, p_n efectúan la secuenciación de los mismos.

Como las secuencias son de n tramas, la ventana de recepción será de $n-1$. Esta situación se ve reflejada en que solo $n-1$ tramas pueden enviarse pendientes de asentimiento, es decir únicamente se podrán disparar las transiciones A_1, \dots, A_{n-1} , mientras no haya una confirmación de la primera trama enviada no podrá ejecutarse A_n .

Cuando las tramas están armadas se transmiten mediante las transiciones B_1, \dots, B_n .

3.2.2 Envío de Tramas

En la fig. 24 se ilustra como una vez que las tramas se emiten se deja indicado que las mismas fueron transmitidas en i_1, i_2, i_3, \dots , pueden dañarse o perderse como lo indica el disparo de P_1, P_2, P_3, \dots , o llegar correctamente a destino ejecutándose L_1, L_2, L_3, \dots . Los lugares a_1, a_2, a_3, \dots señalan el arribo de la trama libre de error.

Con el envío de la trama también se pone en marcha el timer de emisión, el cual controla la transmisión de REQs.

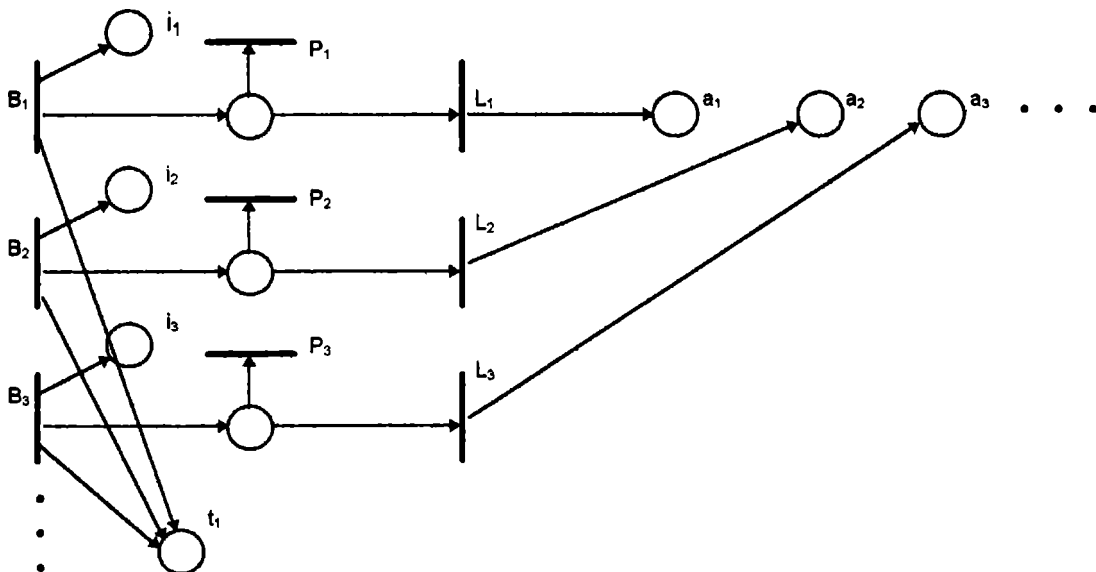


Fig. 24 - Envío de Tramas

3.2.3 Timer de Emisión

Este temporizador está construido internamente con otros, a pesar de que no hay un reseteo de timer por cada trama enviada su comportamiento frente al envío de REQ es idéntico al propuesto. Es decir al vencimiento de la última puesta del timer efectúa el disparo del REQ, siempre y cuando haya habido tramas de información pendientes enviadas con anterioridad.

La fig. 25 señala que al envío de una trama se marca t_1 que dispara S , éste indica en el lugar j que con la puesta del timer se envía una trama. Es decir en este lugar se mantiene la cantidad de tramas emitidas pendientes de confirmación, indicada en número de tokens. El lugar k recibe un token el cual disparará la transición correspondiente, la cual será una de las M_1, M_2, \dots, M_n , permitida por el secuenciador. La razón por la cual se cuenta con n posibles temporizadores y no con $(n-1)$ que a una primera instancia parecería lo más obvio ya que solo pueden enviarse como máximo $(n-1)$ tramas pendientes de confirmación, es que si hubo un timeout y se transmitió REQ, éste mismo pondría en marcha su propio timer, generando con ello la necesidad de otro temporizador.

El disparo de algún M_i generará un token en los lugares q_i, s_i y al indicador de secuencia. Los q_i representan a las tramas, mientras que los s_i tienen un papel interesante ya que son los responsables de dirigir el manejo de los timers.

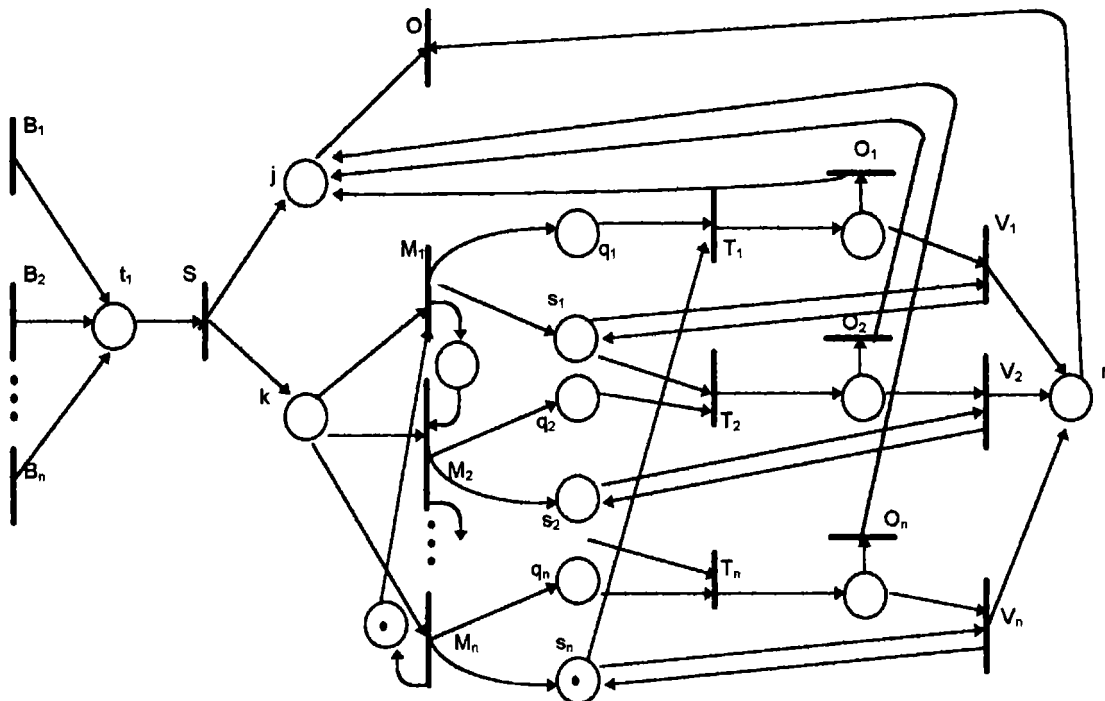


Fig. 25 - Timer de Emisión

Por un lado al llegar una nueva trama tienen la función de deshabilitar el propio y por el otro, en el caso de no haber una nueva trama producirían el seguimiento en curso, pudiéndose en consecuencia generarse un REQ. V_i señala que hubo un timeout del temporizador actual. T_1, T_2, \dots, T_n son los timers propiamente dichos. Un T_i se pone en marcha con la trama y el indicador de su anterior: $s_{i-1} \dots$

Luego del timeout se pondrá fuera de servicio con O_i sacando un token además de j_i , o bien se disparará V_i provocando lo expresado anteriormente.

Una vez producido V_i , se está obligado a considerar si debe o no enviar REQ, es decir si tiene tramas enviadas y éstas no le fueron contestadas. La transición O podría desencadenar un REQ si se satisfacen ciertas condiciones que serán mencionadas en su oportunidad.

3.2.4 Llegada de Tramas y Timer de Recepción

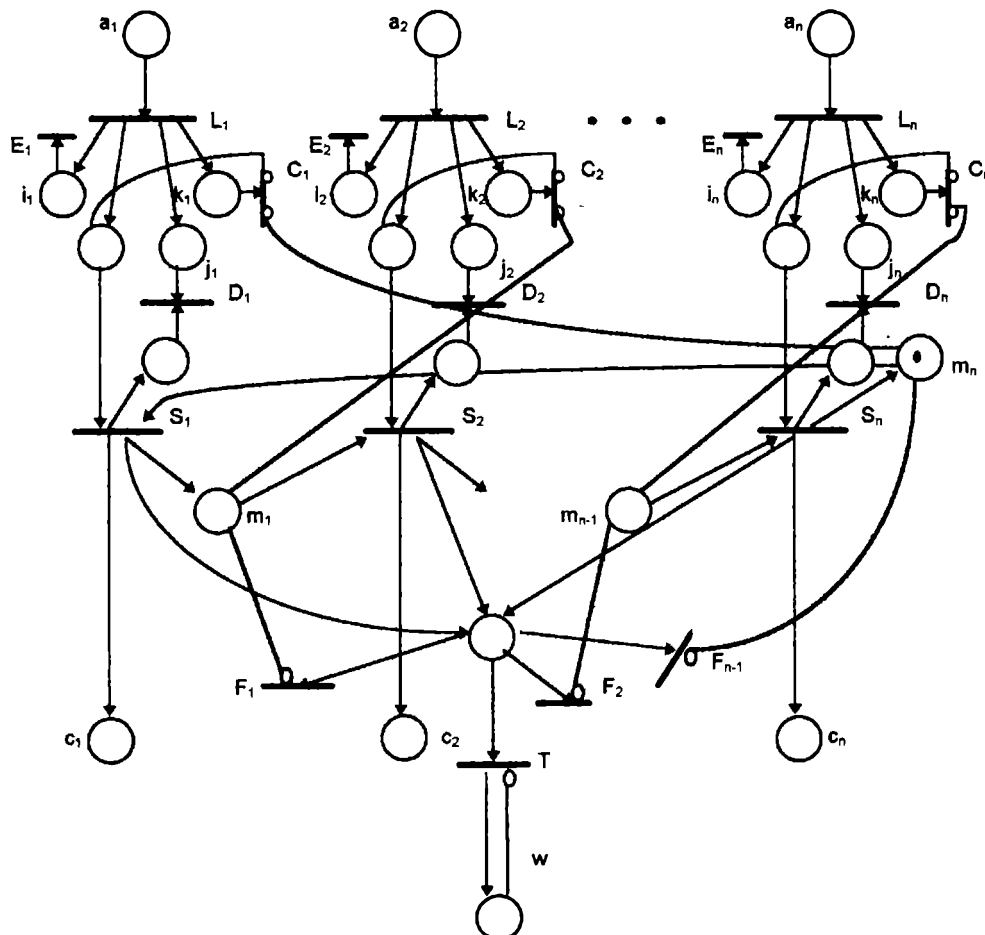


Fig.26 - Llegada de Tramas y Timer de Recepción

Luego que las tramas llegan en secuencia a destino, se procesan y se pone en marcha el timer de recepción, según lo indica la fig. 26.

L_1, L_2, \dots, L_n representan los procesamientos que se realizan ante la llegada de las tramas. Los lugares $i_1, i_2, \dots, i_n, j_1, \dots, j_n, k_1, \dots, k_n$, indican que la trama asociada ha arribado, sus funciones serán expresadas en detalle cuando se trate el pedido de retransmisión.

En este caso juntamente con las transiciones $D_1, D_2, \dots, D_n, E_1, E_2, \dots, E_n$ y C_1, \dots, C_n ejecutan un desempeño tal vez redundante, lo que no quita su necesidad para los fines que se mencionarán con posterioridad.

Los lugares m_i señalan el secuenciamiento de las tramas. Una trama que llega en secuencia tendrá el m_{i-1} (lugar anterior) marcado para que pueda dispararse su propio S_i . Posteriormente a la realización de S_i se da cabida a la siguiente colocando un token en m_i . Simultáneamente con ello, se pondrá en marcha el temporizador del receptor en el caso que no lo estuviera. Cuando llega la primera trama ésta dispara T ya que el lugar w no cuenta con token. Una vez que se encuentra en marcha el timer las próximas tramas no podrán disparar T , por lo tanto harán lo propio con la F_i que les corresponde. El indicador de secuenciación m_i al estar marcado le concede el permiso a F_i .

3.2.5 Confirmación

Luego del vencimiento del timer de recepción se deberán confirmar las tramas llegadas hasta ese momento. Como lo aclara la fig. 27 los lugares marcados en c_1, c_2, \dots, c_n indican las tramas que deben ser asentidas.

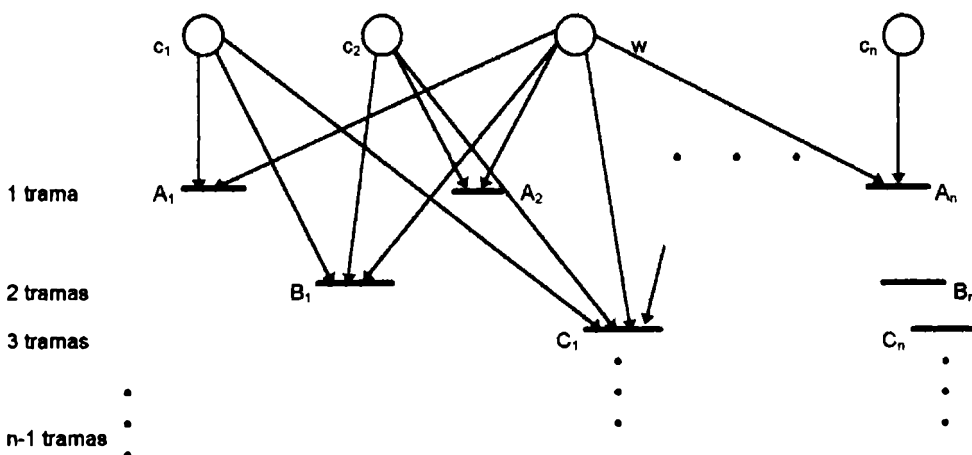


Fig.27 - Confirmación de Tramas

Las transiciones A_1, A_2, \dots, A_n se disparan con la menor prioridad ya que confirman una sola trama. Las que tienen la mayor prioridad son las que confirman $(n-1)$ tramas, si no existen tales se prosigue con las de $n-2$ tramas y así sucesivamente. De este modo se asienten todas las tramas que están habilitadas en c_i .

Como mencionamos anteriormente se cuentan con n transiciones A_i que confirman una trama, lo mismo que para B_i que asienten 2, y así siguiendo. Es decir, existen n transiciones por cada grupo que confirma una cantidad determinada de tramas. Dicho con ejemplos, A_1 es el asentimiento de la trama 0, A_2 de la 1, \dots , A_n de la trama $(n-1)$. B_1 confirma las tramas 0 y 1. B_2 las 1 y 2, \dots , B_n las $(n-1)$ y 0.

3.2.6 Puesta en Línea de la Confirmación

Siguiendo con el mismo criterio usado anteriormente al nombrar con A_i las confirmaciones de una trama, B_i las de 2 tramas, la puesta en línea puede implicar distintas situaciones como se observa en la fig. 28. Los asentimientos como cualquier otra trama pueden llegar intactas o no ser reconocidas al arribar al emisor. No obstante, una de sus ventajas es que pese al hecho de que una confirmación se puede perder, no siempre implica que tendría que retransmitirse la misma. Para ello se cuenta con las transiciones $P_1, P_2, \dots, P_n, Q_1, \dots$. Si bien no llegan a recibirse en el emisor, la trama podría asentirse con la próxima, hecho que se realiza en las transiciones E_1, E_2, \dots . Esta, confluencia la pérdida de la trama anterior con la confirmación actual llevándola al lugar correspondiente.

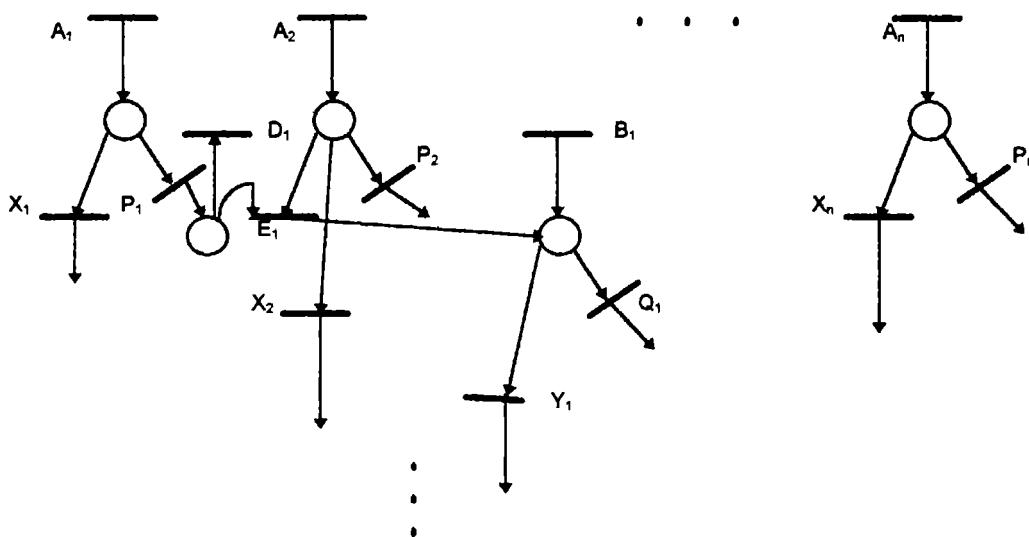


Fig. 28 - Puesta en Línea de la Confirmación

Sin embargo hay asentimientos que pueden perderse definitivamente como ocurre al dispararse D_1, D_2, \dots . Como última instancia la puesta en línea del asentimiento puede alcanzar los propósitos deseados cuando viaja y arriba libre de error, hecho que se ve reflejado en las transiciones $X_1, X_2, \dots, Y_1, \dots$.

Cabe destacar que se explicó solamente la pérdida de una trama. Por razones de espacio se obviaron la otras, aunque el tratamiento con ellas es similar.

3.2.7 Llegada de la Confirmación al Emisor

Con la llegada del asentimiento, el emisor estará capacitado para transmitir nuevas tramas. En la fig. 29 se muestra que H_1 y J_1 son dos de las transiciones que señalan esta situación.

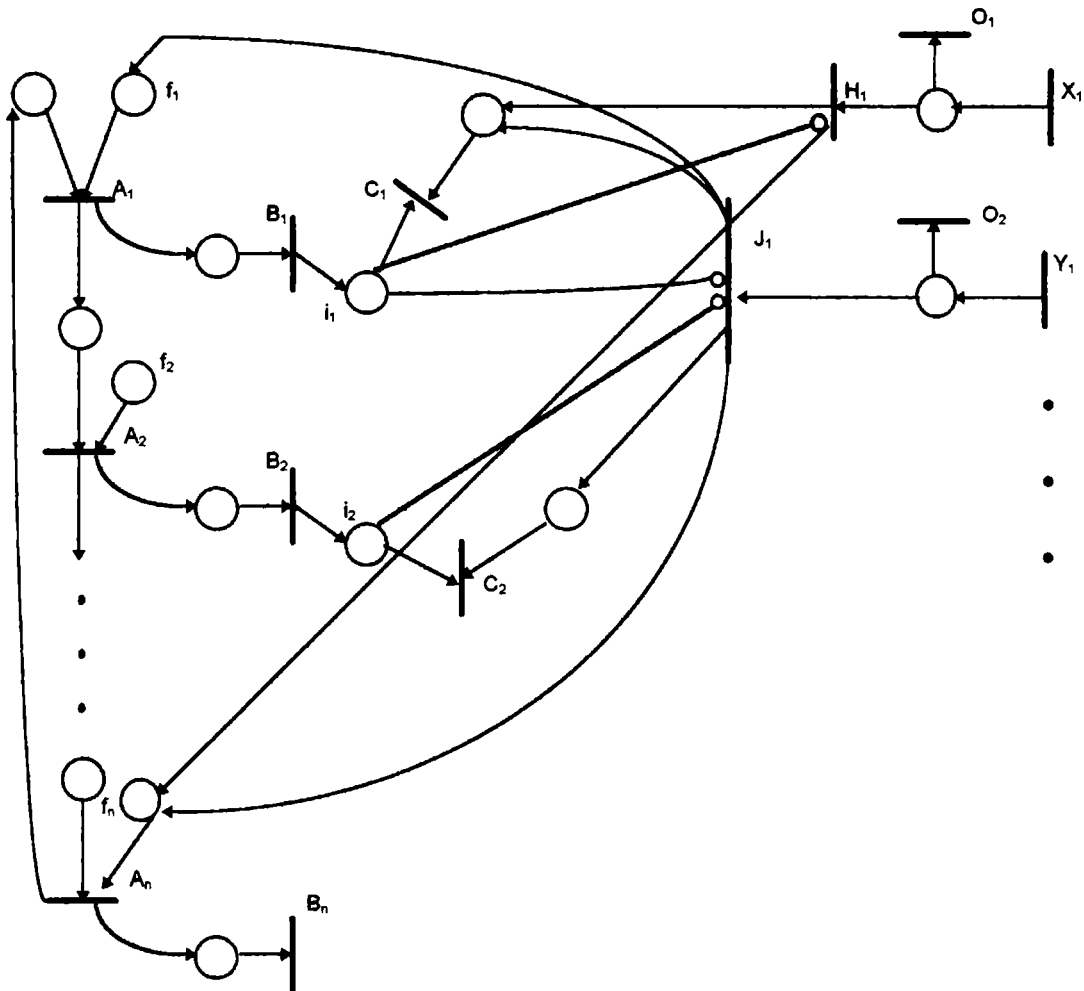


Fig. 29 - Llegada de la Confirmación al Emisor

El manejo es semejante con la confirmación de tres o más tramas. Con respecto a H_1 ésta es disparada si hay token en i_1 , es decir si la misma fue enviada y se halla pendiente de confirmación. Al asentir se dispara C_1 , quitando el token del indicador de envío i_1 a la vez que faculta a A_n para que pueda enviar la trama correspondiente.

J_1 se dispararía si los indicadores i_1, i_2 están marcados. En ese caso vaciaría dichos indicadores para dar paso al envío de dos nuevas tramas, habilitando a las transiciones A_n y A_1 .

3.2.8 Indicadores de Confirmación

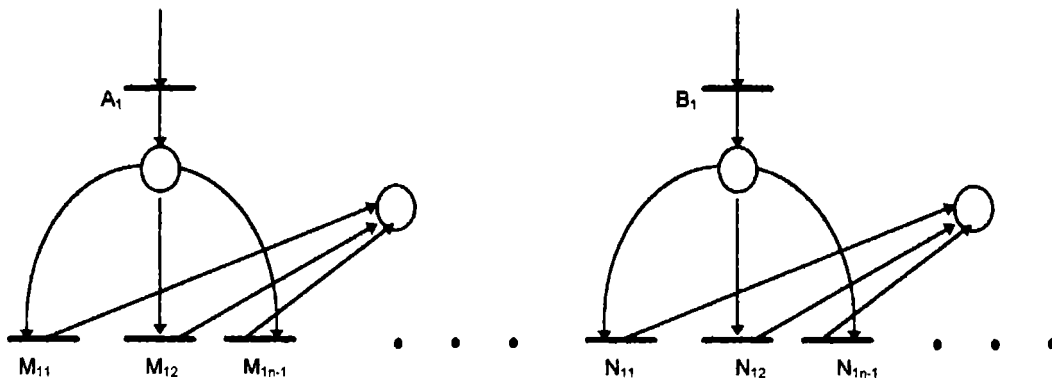


Fig. 30.1 - Indicadores de Confirmación

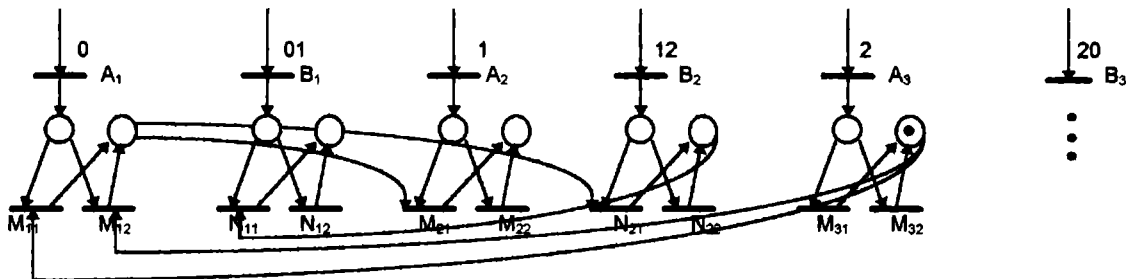


Fig. 30.2 - Indicadores de Confirmación para Tres Tramas

Juntamente con el envío de la confirmación se deja señalado de que asentimiento se trata, el cual será de gran utilidad si la misma se pierde y tiene que ser retransmitida. Para ello como lo indica la fig. 30.1, cada una de las confirmaciones cuenta con $n-1$ transiciones distintas, una de ellas se disparará colocando el token en el lugar correspondiente. Ahora estamos en condiciones

de explicar cual es el mecanismo utilizado, teniendo en cuenta que se debe deshabilitar la confirmación anterior dando cabida a la actual.

Por razones de comodidad se remitirá a un ejemplo, ilustrado en la fig. 30.2 que cuenta con solo tres tramas, en el cual se explicará su funcionamiento.

Suponiendo que ha habido un asentimiento de la trama 0, es decir se disparó la transición A_1 , la próxima que podrá habilitarse sería M_{11} , al ejecutarse la misma dejaría sin efecto la confirmación anterior señalando en el indicador correspondiente el presente asentimiento.

Posteriormente, las posibles confirmaciones que podrían ejecutarse serían las pertenecientes a las tramas 1 o las 1 y 2. De acuerdo con ello salen del lugar indicador dos arcos, uno que se dirige a la transición M_{21} y otro a N_{21} , para que pueda dispararse una de ellas.

3.2.9 Pedido de Retransmisión

Hasta el momento se trataron situaciones donde el flujo de información se daba normalmente, si bien se mencionó que podrían ocurrir pérdidas o deterioros en las tramas.

Un pedido de retransmisión ocurre cuando se presenta en el receptor una nueva trama sin que lo haya hecho su precedente.

En un principio se especificará la solicitud para una sola trama para luego dilucidar el pedido de más tramas consecutivas.

En la fig. 31.1, presumiendo que llegó una trama al lugar a_2 se realiza L_2 . Al no haber arribado la anterior no se producirá S_2 frenando el normal desenvolvimiento. En su lugar se dispara R_1 ya que tanto el lugar del indicador de secuencia como el j_1 no tienen marcas, con ello se hace efectivo el pedido de retransmisión para la trama ausente. La razón por la cual se testea en ambos lugares y no solo en el indicador de secuencia como parecería lo mas evidente, es que si la trama llegó y se está ejecutando S_1 todavía no habría un marcado para el señalador, con lo cual se provocaría un pedido indeseado. Verificando en los dos lugares este hecho se ve solucionado. En el lugar l_1 indica que la misma fue solicitada al emisor. Posteriormente al reenvío y el arribo se ejecuta G_1 señalando de este modo que la operación ha finalizado.

Nótese que si hubiese llegado otra trama posterior a la que hace la petición, no podía originar ninguna acción, mientras no arribe la requerida.

El pedido de retransmisión para más de una trama es equivalente al anterior. Se ejemplifica con el pedido de dos tramas ilustrado en la fig. 31.2.

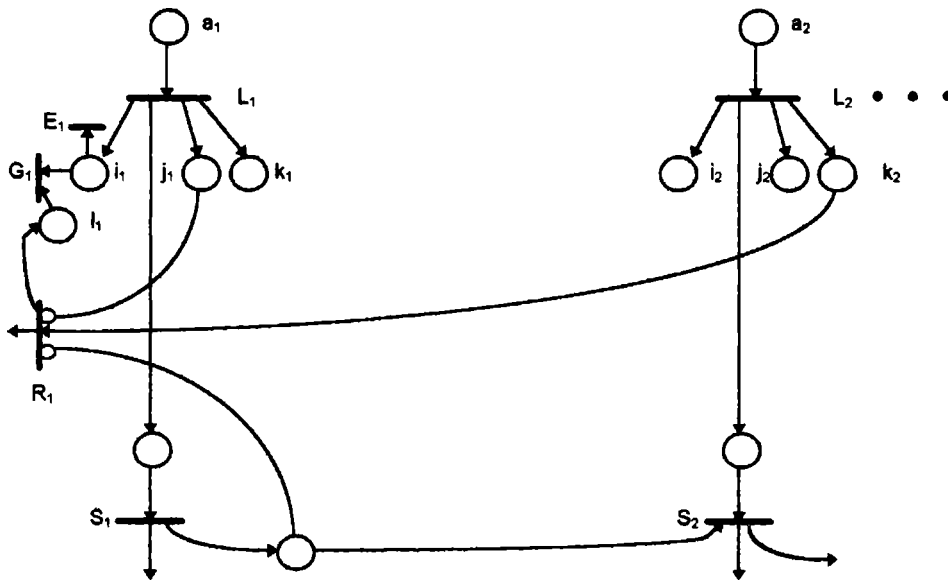


Fig. 31.1 - Pedido de Retransmisión para una Trama

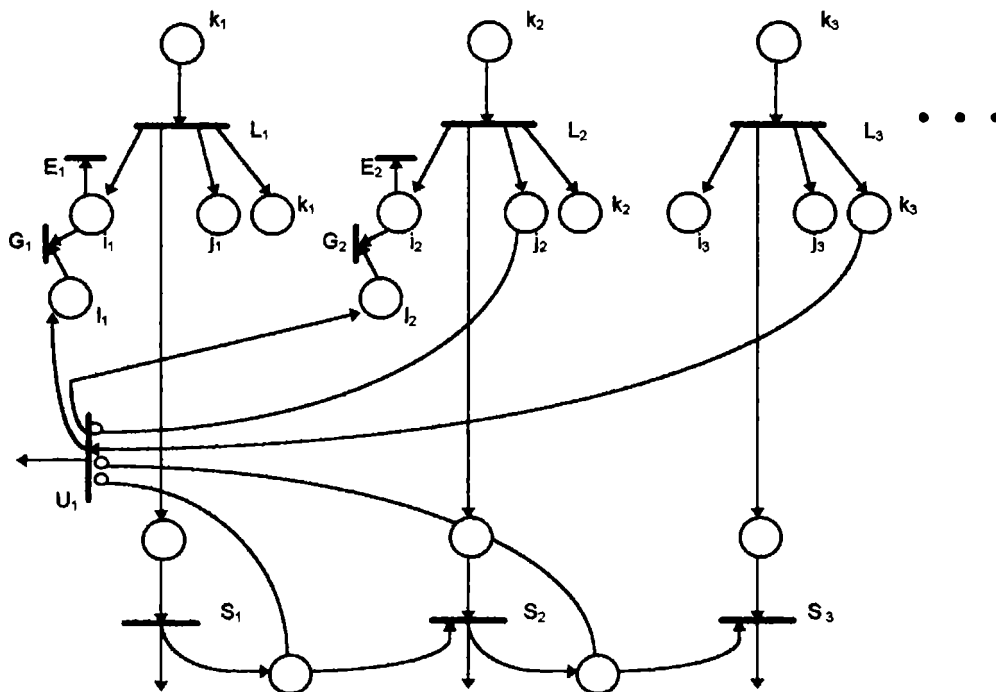


Fig. 31.2 - Pedido de Retransmisión para Dos Tramas

La transición U_1 es disparada chequeando los dos indicadores anteriores y el lugar j_2 . Como resultado almacena sendos tokens en los lugares l_1 y l_2 , que serán quitados al arribo de sus respectivas tramas por G_1 y G_2 .

Una nueva pérdida de la primera trama solicitada de las dos, pero llegada de la segunda implicaría otro pedido, impulsado esta vez por esta última trama.

Finalmente, se explicarán las transiciones C_i , D_i y E_i correspondientes a la fig. 26 - Llegada de Tramas y Timer de Recepción, que se dejaron en suspenso. Las E_i se disparan siempre y cuando llega una trama, que no haya sido requerida con anterioridad. Las D_i son auxiliares que se disparan luego que la trama ha arribado y está a la espera del timeout para el asentimiento. Las C_i son otras que cooperan, asegurando que la trama vino en secuencia.

3.2.10 Pedido de Retransmisión en Línea y Llegada

Luego de armar el pedido de retransmisión se lo transmite según se esclarece mediante la fig. 32.1. El mismo puede sufrir daños en el transcurso, como ocurre al dispararse K_i o llegar ileso. En este caso, se ejecutarían sucesivamente J_i y M_i . Las razones por la que pasan sucesivamente por ambas se mencionarán oportunamente al tratarse la retransmisión del pedido ante un REQ. M_i produce que se desagote el indicador de trama i_i observando que la misma necesita reenviarse.

Si el pedido implicara más de una trama, tendría tantas transiciones como tramas para disparar hasta llegar a la última. En el ejemplo de la fig. 32.2 se muestra el pedido de dos tramas. El motivo por el cual se utilizan tantas transiciones se debe a que las emisiones deben ser secuenciales y no simultáneas.

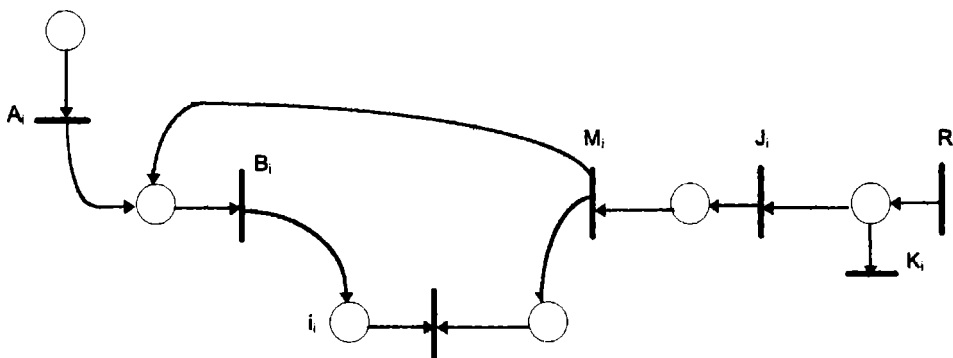


Fig. 32.1 - Pedido de Retransmisión para una Trama

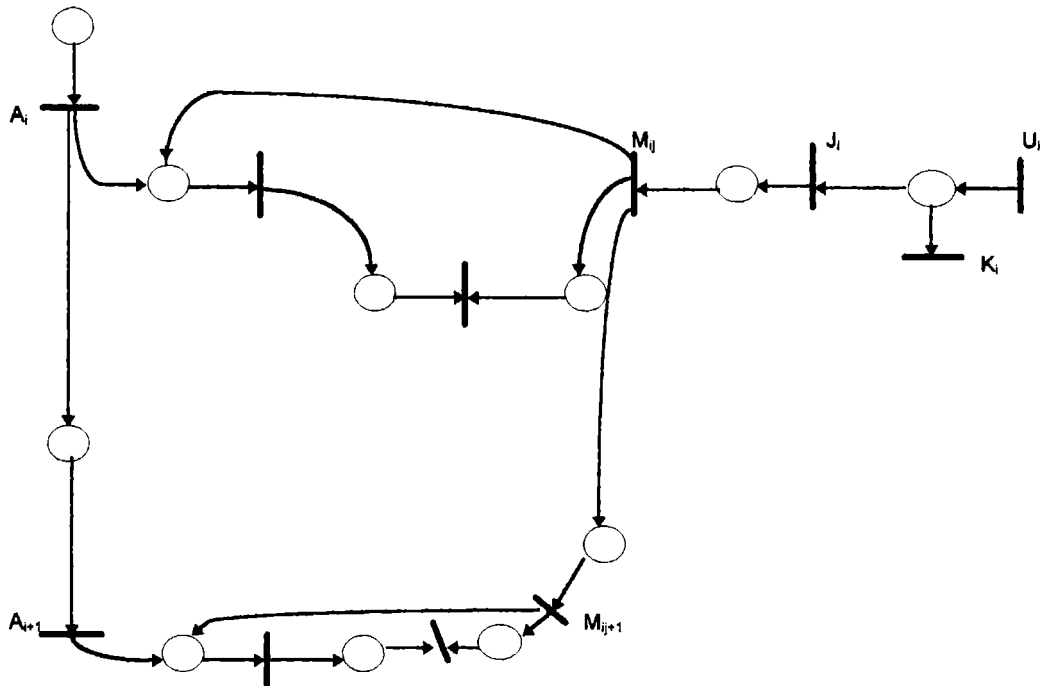


Fig. 32.2 - Pedido de Retransmisión para Dos Tramas

3.2.11 Envío del REQ

Debido a que no se pueden solucionar todos los problemas con el pedido de retransmisión, se debe recurrir a la emisión del REQ. La misma debe indagar al receptor y así obtener la respuesta adecuada para tomar una resolución.

Un REQ se envía luego del timeout, teniendo en cuenta si tiene enviadas tramas de información de las cuales no se conoce que les ha ocurrido. Se explicarán los mecanismos en base a la fig. 33. Para ello debe disparar la correspondiente transición P_i . Las mismas tienen un orden de prioridad en el disparo. Es decir si el problema es por ejemplo con las tramas 0 y 1, las cuales están señaladas con sendos indicadores de envío i_i , ambas P_1 y P_2 podrán realizarse. No obstante, la tarea debe ser realizada por P_2 ya que es prioritaria ante P_1 , en otras palabras no solo se desconoce lo que le ha sucedido a la trama 0, sino también a la 1. Además para que el disparo sea efectivo no deberá haber tramas enviadas con posterioridad indicadas en el lugar j , si las hubiere se ejecutaría Q dejando sin efecto el REQ. El disparo del REQ significaría su transmisión, pero además la disposición para arrancar nuevamente el timer, acto que se efectúa mediante la

transición R. También otra consecuencia de la ejecución de P_i es dejar marcado el lugar v señalando con ello el envío de la trama respectiva.

La reinicialización del temporizador se realiza de un modo análogo que para una trama de información.

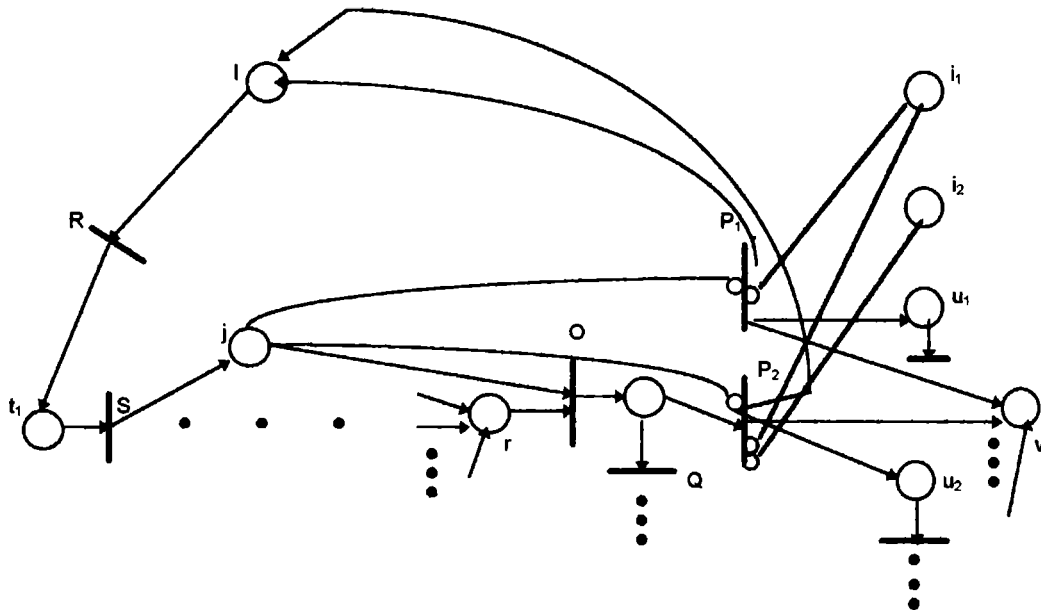


Fig. 33 - Envío de REQ

3.2.12 Respuesta ante un REQ

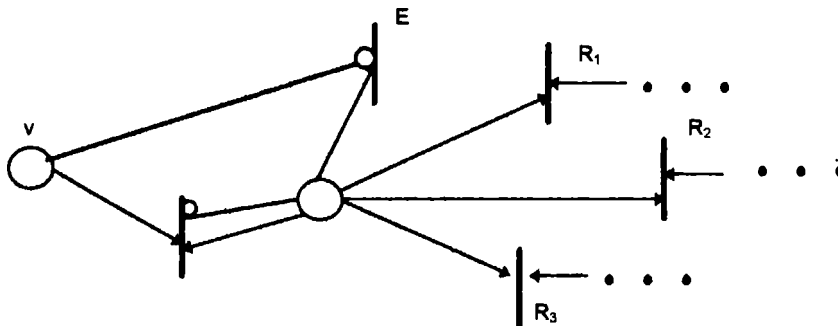


Fig. 34 - Respuestas ante el REQ

Luego de ser transmitido un REQ, éste podrá arribar al receptor o sucumbir ante el ruido existente en el medio. Ante este último caso, vencerá nuevamente el timer del emisor generando otro REQ.

La reemisión de un nuevo REQ significará que deberá sacrificarse el token anterior extrayéndolo a través de E según se observa en la fig. 34, para dar entrada al nuevo indicador.

El hecho de que el REQ sea reconocido por el destinatario puede implicar diferentes respuestas. Con este fin se creó el indicador, o sea al ser contestado deshabilitaría el REQ. R_1 señala un pedido de retransmisión, R_2 una confirmación y R_3 un asentimiento anterior que involucraría la retransmisión de tramas de información. Las transiciones precedentes son las posibles contestaciones al REQ.

A continuación se explicarán en detalle, las consecuencias del envío de REQs.

3.2.12.1 Pedido de Retransmisión

El protocolo prioriza el pedido de retransmisión a las confirmaciones, por lo tanto se generan un orden de primacías de las posibles contestaciones.

Como primera medida debe verificar si hay algún pedido de retransmisión que quedó pendiente.

Así se ilustra en la fig. 35, la llegada del REQ indicada en el lugar u_i dispara la transición O_i . Como se expresó precedentemente al enviarse un pedido de retransmisión se indica en I_i , mientras esa trama no se presente, ese lugar estaría marcado. En consecuencia permitiría el disparo de O_i , determinando una nueva solicitud. La misma se realiza análogamente a un pedido común. Sin embargo, ante la llegada de la trama del pedido al emisor se dispara R_1 quedando con ello finalizada la gestión del REQ. En los casos normales se ejecuta S_i . Al dispararse R_1 , deja un token en el lugar n .

Esta marca cumple un papel de singular importancia cuando sucede un solapamiento de tiempos. En un instante siguiente a la transmisión del REQ se observa un pedido de retransmisión. El emisor la toma como una respuesta válida disparando R_1 , pero el REQ sigue su camino en el receptor activando otro pedido con O_i . Luego la solicitud avanzaría hacia el emisor, sin embargo no la toma en cuenta disparando N_i , ya que dispone de un token en el lugar n .

Ante una pérdida del pedido frente al REQ o de él mismo, hay un nuevo vencimiento del timer que dispara otro REQ, quitándose la marca de n , al realizarse Q .

Si el proceso sigue un curso normal, es decir que enviado REQ responde con un pedido y la trama de información retransmitida arriba correctamente al destinatario, se dispara G_i quedando

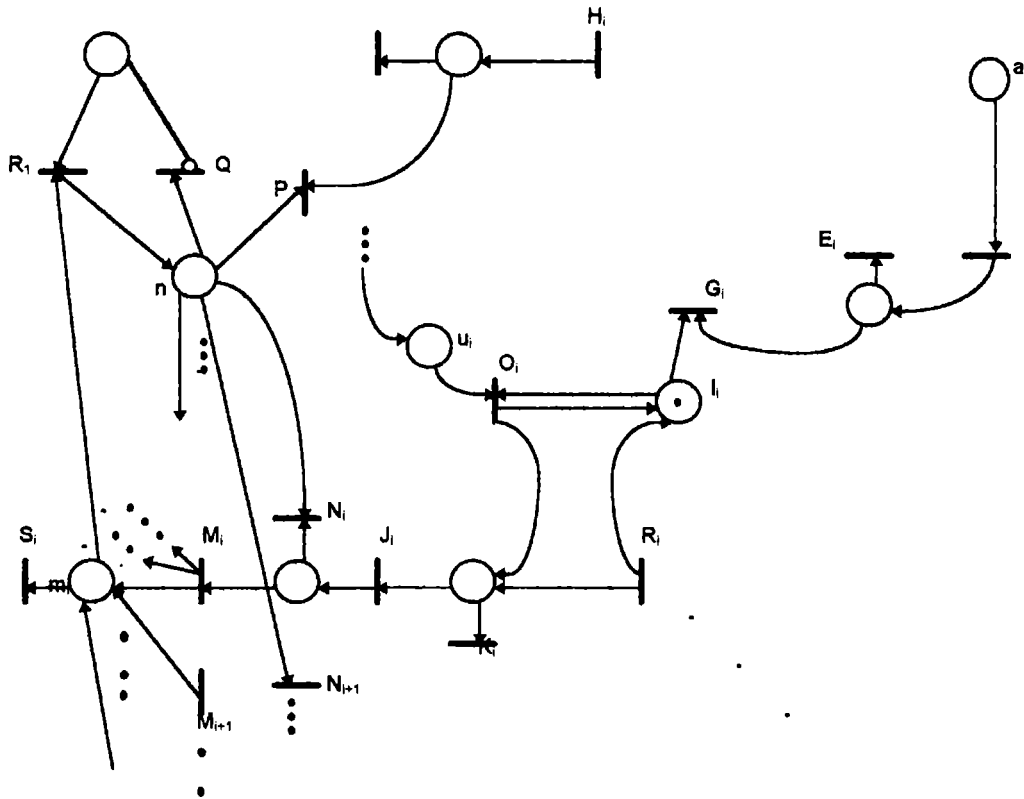


Fig. 35 - Pedido de Retransmisión ante un REQ

con esto concluida la operación del pedido. No obstante, quedó un token en n que será quitado al haber una confirmación mediante la transición P .

3.2.12.2 Confirmación

Si el indicador del pedido de retransmisión no está habilitado, el REQ debe testear si hubo confirmación anterior referente a las tramas que tiene pendientes.

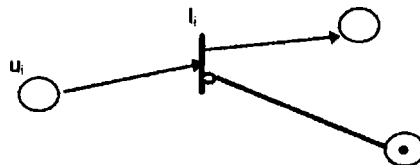


Fig. 36 - Confirmación ante el REQ

La fig. 36 muestra la transición I_i que se ejecuta cuando el indicador de confirmación tiene el correspondiente token. REQ disparará la misma originando una nueva confirmación.

Una trama de confirmación que fue dañada por el ruido existente en la línea puede ser salvada de esta manera.

Un problema que podría suscitarse consiste en los solapamientos de los tiempos de REQ y de confirmación. Es decir sucedería un hecho similar al narrado anteriormente con el pedido de retransmisión. Como en ese caso se toma como válida la primera confirmación. La segunda generada por el REQ deberá ignorarse. Para ello en la fig. 29 - Llegada de la Confirmación al Emisor, se testea antes que las tramas se hayan enviado, para luego dar cabida a la transmisión de tramas nuevas, es decir para el corrimiento de la ventana. Retomando lo expresado, cuando la segunda confirmación arriba al emisor ya no contará con que las tramas fueron enviadas, según se observa en la fig. 37 en consecuencia se disparará O_i que significará una confirmación pero que no modifica la ventana.

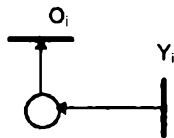


Fig. 15 - Nueva Confirmación Arribada

La confirmación como respuesta al REQ se ve manifestada al dispararse R_2 , mientras que con un asentimiento común se ejecuta Z según lo indica la fig. 38.

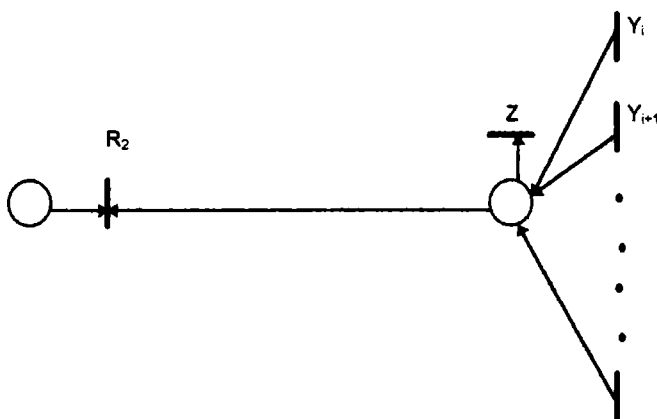


Fig. 38 - Confirmación como Respuesta al REQ

3.2.12.3 Retransmisión de Tramas

Si el indicador de confirmación no estuviera marcado, como así tampoco ninguna de los grupos con menor prioridad anteriores a la que pertenecen esas tramas se disparará la transición respectiva que efectúa la retransmisión. Pero no siempre tendrá como resultado el reenvío de tramas, ya si se hayan enviado tramas posteriores, significaría una confirmación anterior, lo que no modificaría en absoluto al emisor. Ante una situación de este tipo sólo tendría efecto el pedido de retransmisión, ya que una confirmación podría generar que se reenvíen tramas no deseadas.

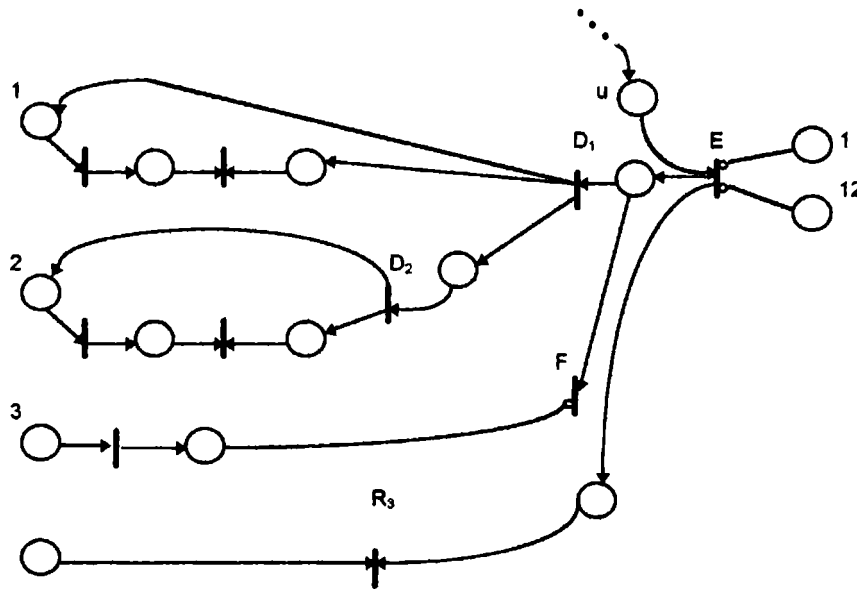


Fig. 39 - Retransmisión de Tramas ante un REQ

Sea por ejemplo, en la fig. 39 que los indicadores de las tramas 12 y 1 se encuentren vacíos, lo que implica que estas dos tramas se perdieron o se dañaron. Entonces actuará en base a lo siguiente. Primero chequea si hay tramas subsiguientes enviadas testeando el indicador de emisión, ante una respuesta afirmativa se dispara F significando que no hay movimiento de ventana. Si F no puede ejecutarse se dispara D₁ con la consiguiente retransmisión de la trama 0 y D₂ con la de la 1. Al igual que con el pedido de retransmisión, se efectúa en etapas para no perder el secuenciamiento.

Paralelamente con la ejecución de una de estas dos tareas se debe responder al REQ, acción que se realiza con el disparo de R₃.

3.2.13 Temporizaciones

Se deja indicado que los timeouts deben darse a tiempos coherentes, no permitiendo vencimientos prematuros, lo que provocaría por parte del emisor el disparo de REQ, mientras la confirmación o el pedido de retransmisión está aún viajando. Cabe destacar que si bien se consideraron estos aspectos para la definición del protocolo como para la RdP, no es un signo deseable. Del mismo modo, un receptor al que su timer se le vence continuamente, no optimizaría en absoluto la transmisión.

De acuerdo a lo expresado previamente para el diseño del timer del emisor se utilizaron n timers internos, uno para cada una de las tramas que podrían enviarse pendientes de confirmación, es decir un total de $(n-1)$ además de otro que sería utilizado por el REQ. El tiempo de cada uno se eligió contemplando la llegada de la confirmación.

3.2.14 Piggybacking

Los aspectos del protocolo mencionados sobre la RdP hasta el momento no involucraban el piggybacking. La idea del mismo consiste en superponer la confirmación con una trama de información. El envío de la misma deshabilitaría el timer de recepción.

En la fig. 40 se describirá a grandes rasgos el comportamiento del piggybacking abstrayendo funciones. El conjunto de confirmaciones se denominará con las transiciones $H_1, \dots, H_{n(n-1)}$. Una trama con el asentimiento superpuesto disparará su respectivo H_{ij} , indicando con ello por un lado el envío de ella y por el otro la confirmación.

Sin embargo si la Capa de Red no le envía paquetes a su inmediata inferior, vencerá el timer de recepción ejecutándose el correspondiente H_{Tj} que provocará el mismo asentimiento.

Cabe acotar que se describió el funcionamiento para un solo nodo. El otro constará con los mismos componentes que a su vez emitirá sus propias tramas de información superponiendo las confirmaciones para las recibidas de este extremo.

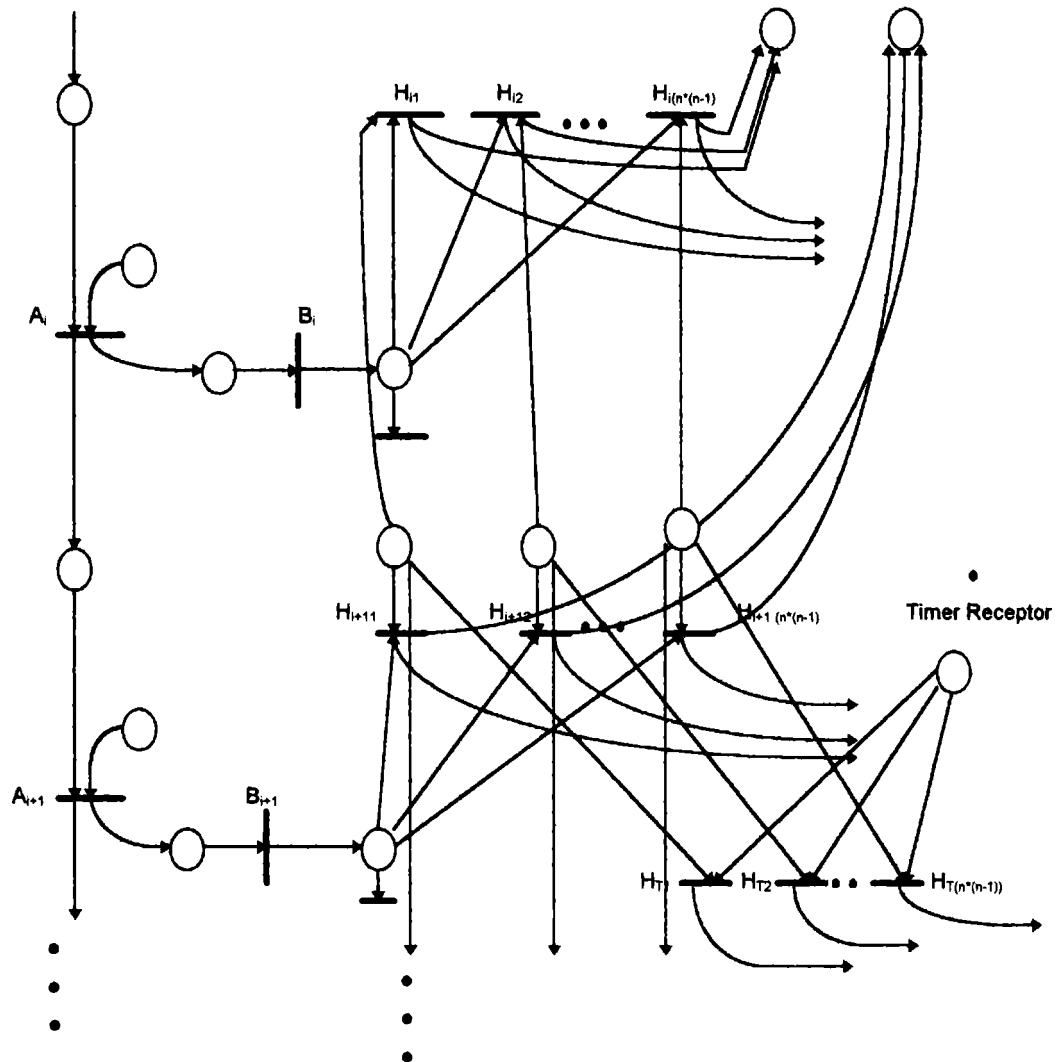


Fig. 40 - Piggybacking

3.3 Simulación

Una vez descrita la RdP omitiendo el piggybacking se volcó la misma a un simulador denominado VISUAL SIMNET v1.35, extraído de la dirección FTP anónimo: <ftp://inf.tu-dresden.de/pub/ms-dos/simulate/simnt135.zip> de Internet.

Esta herramienta brinda un ambiente de procesamiento que posibilita visualizar la secuencia de ejecución de la RdP animada y genera un archivo con dicha corrida. Permite medir



FACULTAD
DE INGENIERÍA
U.N.L.P.

los tiempos de servicio, tiempos de actividad y de ocio, además de mostrar la cantidad de tokens de un lugar en un momento determinado.

Para los tiempos de las transiciones se utilizaron las distribuciones normales para aquellos que requerían de un tiempo de servicio y para otros, entre ellos los timers la distribución constante.

La eficiencia de la línea para los términos de simulación se determinó en 0.9. Se tuvieron en cuenta las pérdidas individuales y considerando que el ruido generalmente se da en ráfagas, como cuando se extravían todas las tramas de la ventana emisora provocando el envío de REQ, hechos que se ven reflejados en la simulación.

Por motivos de comodidad y mejor entendimiento se modelizó una RdP con tres tramas, ventana dos. No obstante podrá ser modificada, debido a que los mecanismos utilizados son los mismos, como se mencionó anteriormente. Cabe destacar que el crecimiento de lugares y transiciones aumenta con progresión geométrica.

Capítulo IV

CONCLUSIONES FINALES

4. Conclusiones Finales

El objetivo del proyecto fue definir un protocolo optimizado que brinde un desempeño eficiente en un medio que introduce errores en la transmisión y con retardos. A tal fin se profundizó el estudio de los protocolos standards, centrando el interés en las tres primeras capas del modelo OSI, en especial al nivel de Enlace.

Un segundo objetivo fue estudiar Redes de Petri como herramienta de simulación de protocolos, la cual nos permitió ahondar en detalles, al igual que descubrir y corregir errores en el desarrollo de las propuestas. Las características propias de los protocolos permite su modelación a través de dicha herramienta, recurso que generalmente no se encuentran dentro de los lenguajes de programación.

Como complemento a la descripción efectuada mediante Redes de Petri, se generaron los algoritmos de las propuestas desarrolladas. Del diseño de los mismos se desprende que el tiempo de CPU utilizado para ambas propuestas es diferente. La propuesta I, al estar testeando todos los timers de emisión implica un tiempo extra de procesamiento, hecho que no sucede en la propuesta II, debido a que tiene que controlar un único temporizador. Otro aspecto a tener en cuenta es que en la propuesta I se utiliza más memoria que en la otra, al tener un procedimiento que controla los timers por cada trama emitida. Además los vencimientos de los timers originará interrupciones en el flujo normal del programa principal. Si las tramas fueron emitidas en intervalos de tiempos muy cortos, los vencimientos se producirán en cadena si nos circunscribimos a la propuesta I. En la propuesta II se reduce a una única interrupción por existir un único timer.

Si bien las dos propuestas enfatizan las funciones y servicios que debe realizar la Capa de Enlace, en la actualidad existen protocolos en donde el control de errores se lleva a cabo en capas superiores. Esta idea se basa en la alta confiabilidad de los enlaces, tal es el caso del Frame Relay que elimina una capa completa del modelo OSI.

Sin embargo, en algunos medios de transmisión los retardos son importantes, por ejemplo enlaces satelitales, si a esto se le anexa el ruido que puede existir en el medio, el hecho de retransmitir tramas habiendo llegado correctamente al receptor afecta la performance del enlace.

La solución a este tipo de planteo lo brindan las dos propuestas descriptas.

Las dificultades que se presentaron en la formulación del protocolo fue en la descripción minuciosa de todas las situaciones distintas que se generan en la transmisión de datos. Bajo ésta instancia la generación de un protocolo debe contemplar todos y cada uno de éstos ítems y dar una solución óptima a los planteos. A tal fin la utilización de las redes de Petri contribuyó a delinear soluciones.

En cuanto a la generación de la red de Petri, las dificultades que se plantearon giraron en torno a su implementación. Con lo cual se requiere un análisis top down del problema, modularizar funciones y de acuerdo a ello volcarlo a la red.

El proyecto, dado su grado de especificación, es punto de partida para su implementación. Los protocolos pueden ser implementados en instalaciones computacionales posiblemente heterogéneas. Bajo esta premisa dichas implementaciones pueden ir modificándose de acuerdo al tipo de instalación en el que se encuentran. El objetivo de esta fase es obtener el protocolo efectivamente en ejecución.

Una vez implementado el protocolo debe ser verificado de acuerdo a su conformidad en relación a la especificación. Para que este testeo sea realizado existe un itinerario a ser seguido que describe las acciones a ser tomadas y las respuesta que deben ocurrir. Este itinerario puede ser realizado en base a una correcta especificación de protocolos.

Finalmente podemos decir que la especificación desarrollada puede ser guía de futuros protocolos.

Capítulo V

BIBLIOGRAFÍA

5. Bibliografía

Andrews, Gregory R.: " Concurrent Programming. Principles and Practice.", The Benjamin/Cummings Publishing Company, Inc., 1991.

Pasaje de mensajes. Pasaje de mensajes asincrónicos. Pasaje de mensajes sincrónicos.

Girault, Claude y Reisig, Wolfgang: " Application and Theory on Application and Theory of Petri Nets.", Springer-Verlag Berlín Heidelberg, 1982.

Aplicación de redes a sistemas de tiempo real. Modelación y alcanzabilidad de Redes de Petri en algoritmos distribuidos.

Hoare, C. A. R.: "Communication Sequential Processes.", Prentice Hall, 1985.

Concurrencia de procesos. Comunicación entre procesos.

IBM: "Synchronous Data Link Control. Concepts.", 1986.

Silva, Manuel: "Las Redes de Petri: en la Automática y la Informática.", AC, 1982

Introducción a las Redes de Petri como herramienta de descripción funcional de sistemas secuenciales y concurrentes. Formalización y aplicación a la modelación de sistemas.

Silva Nogueira, José Marcos: "Protocolos e Serviços de Comunicação: Principios, Especificação e Teste.", V EBAI, 1991.

Ciclo de vida de los protocolos. Especificación de protocolos y servicios. Técnicas de descripción formal.

Silva Nogueira, José Marcos y De Castilho Lages, Newton Alberto: " Introdução aos Sistemas Distribuidos", Unicamp, 1986.

Las tres primeras capas del modelo OSI : protocolos y servicios. Subsistema de interconexión.

Sloman, M. S.: "Standards and Protocols. X.25 Explained.", IPC Business Press, 1978.

Stallings, William: "ISDN and Broadband ISDN.", Macmillan, 1992.

Nuevas tecnologías y tendencias en protocolos. ATM, Frame Relay.

Stallings, William: "Data and Computer Communications.", Macmillan, 1985.

Tanenbaum, Andrew S.: "Redes de Ordenadores.", Prentice Hall Hispanoamericana, 1991.

Tanenbaum, Andrew S.: "Sistemas Operativos Modernos.", Prentice Hall Hispanoamericana, 1992.
Conceptos básicos sobre sistemas operativos distribuidos.

Tanenbaum, Andrew S.: "Sistemas Operativos Distribuidos.", Prentice Hall Hispanoamericana, 1985.

Concepto de sistemas operativos distribuidos. Proceso de comunicación en los sistemas distribuidos. Bloqueo, detección y prevención.

Capítulo VI

GLOSARIO

6. Glosario

- **ancho de banda** : el espectro de una señal es el rango de frecuencias que lo contiene. El ancho de banda absoluto de una señal es el ancho del espectro. Sin embargo, la mayor parte de la energía de la señal está incluida en una banda relativamente angosta de frecuencias. Esta banda se denomina ancho de banda efectiva o solamente ancho de banda.
- **comando** : información que fluye del primario al secundario.
- **configuración balanceada** : usada en conexiones punto a punto. Consiste de dos estaciones combinadas y soporta transmisión half-duplex y full-duplex.
- **confirmación desbalanceada** : usada en conexiones punto a punto y en multipunto. Consiste de un primario y una o más estaciones secundarias y soporta transmisión half-duplex y full-duplex.
- **DCE** : (Data Circuit-terminating Equipment), equipo usado para establecer, mantener y terminar la conexión y provee una modulación apropiada de la señal para la transmisión.
- **disciplina orientada al bit** : no utiliza caracteres de control como delimitadores, sino que se define una estructura fija, la trama tiene una longitud fija.
- **disciplina orientada al carácter** : usan caracteres de control de un conjunto de códigos dados para delimitar tramas y para el intercambio de datos de control. Hay dos grandes categorías : la primera limita una trama por caracteres de control especificados al comienzo y al final. La segunda los establece mediante el número de caracteres o bytes de datos que la trama contiene.
- **DTE** : (Data Terminal Equipment), equipos tales como procesadores, controladores y terminales que se comunican mediante un enlace de datos.
- **enlace multipunto** : se caracteriza porque más de dos dispositivos comparten el medio de transmisión.
- **estación combinada** : combina las características de las estaciones primaria y secundaria. Una estación combinada puede utilizar comandos y respuestas.
- **enlace punto a punto** : se caracteriza porque existe un enlace directo entre dos dispositivos que comparten un medio de transmisión.
- **estación primaria** : en una conexión, un nodo es responsable de controlar la operación de enlace.
- **estación secundaria** : opera bajo el control de la estación primaria. Los frames usados por la(s) estación(es) son llamadas respuestas. El primario mantiene un enlace lógico con cada estación primaria en línea.
- **full-duplex** : configuración que permite que dos estaciones implicadas transmitan simultáneamente.

- **half-duplex** : configuración que posibilita a dos estaciones transmitir en ambas direcciones, pero sólo una lo lleva a cabo por vez.
- **LS** : (Link Station) es el hardware y software que permite la conexión y proporciona el control del enlace.
- **modelo OSI** : (Open System Interconnection) especifica niveles y aspectos de la comunicación, determina nombres estandarizados y señala el trabajo que realiza cada nivel. Fue desarrollado por la ISO : International Standards Organization.
- **modo de transferencia ABM** : (Asynchronous Balanced Mode) es una configuración balanceada. Cualquiera de las estaciones combinadas puede iniciar la transmisión sin permiso explícito de la otra estación combinada.
- **modo de transferencia ARM** : (Asynchronous Response Mode) es una configuración desbalanceada. El secundario puede iniciar la transmisión sin permiso explícito del primario (por ejemplo enviar una respuesta sin esperar el comando). El primario tiene la responsabilidad de la línea, incluyendo inicialización, recuperación de errores y desconexión lógica.
- **modo de transferencia NRM** : (Normal Response Mode) es una configuración desbalanceada. El primario puede iniciar la transferencia de datos hacia un secundario, pero un secundario puede sólo transmitir datos en respuesta a un poll enviado por el primario.
- **protocolo** : conjunto de reglas que gobiernan el formato y significado de los mensajes que son intercambiados entre entidades en diferentes sistemas. Las entidades utilizan protocolos para definir servicios.
- **respuesta** : es la información que fluye del secundario al primario a modo de contestación por algún comando o información enviada por el primario.
- **ruido** : es toda señal no deseada que se inserta en la señal transmitida durante el momento de la transmisión y que forma parte de la señal recibida.
- **servicio** : conjunto de operaciones que proporciona una capa a la capa superior. El servicio define las operaciones que la capa efectúa. Un servicio se refiere a una interfase entre dos capas, siendo la inferior la que lo provee la superior la que lo utiliza.
- **servicio con conexión** : el emisor y el receptor establecen primero explícitamente una conexión y negocian el protocolo a utilizar.
- **servicio sin conexión** : no requiere de una configuración previa del enlace. El emisor transmite el primer mensaje cuando está listo para hacerlo.
- **transmisión asíncronica** : es también conocida como transmisión start-stop, constituye una solución para el problema de sincronización, efectuando el envío de pequeños paquetes de bits. Está orientada al carácter y se usa entre 5 y 8 bits como datos.
- **transmisión sincrónica** : se utiliza sincronización entre emisor y receptor a través de relojes. Existe un delimitador al comienzo y al final de los datos. Los datos van unidos a la información de control formando un frame.

- **ventana** : es un mecanismo de control de flujo usado por cada canal lógico a través de la interfase. La ventana es el número de paquetes no confirmados que el DTE o DCE puede tener pendientes en un instante de tiempo sobre un canal lógico particular.
- **ventana de emisión** : lo constituye la cantidad de tramas que el emisor puede enviar sin que sean confirmadas.
- **ventana de recepción** : es la cantidad de tramas que el receptor está autorizado a aceptar.

Apéndice A

RESULTADOS DE LA SIMULACIÓN

A. Resultados

Se tomaron muestras de corridas diferentes del TESIS.SIM adjuntado, con el objeto de analizar los casos que se iban sucediendo. Para los ejemplos 2.2 y 7 se construyeron redes de Petri diferentes que permitiesen mostrar los casos.

Algunos de los ejemplos más notorios serán dilucidados a continuación. Por motivos de espacio se simplifican los comentarios añadidos a los ejemplos de las corridas, las que se tomaron en crudo y no revisten ninguna modificación. Únicamente serán resaltadas las transiciones que efectúan una tarea importante, no así las secundarias. Posteriormente se describirán con mayor detalle. Las abreviaciones T1, T2 y T3 indican los tres temporizadores que simulan el timer de emisión. TR es el perteneciente al receptor. t0, t1 y t2 son las tramas 0, 1 y 2 respectivamente.

Las salidas consisten de cinco columnas que indican:

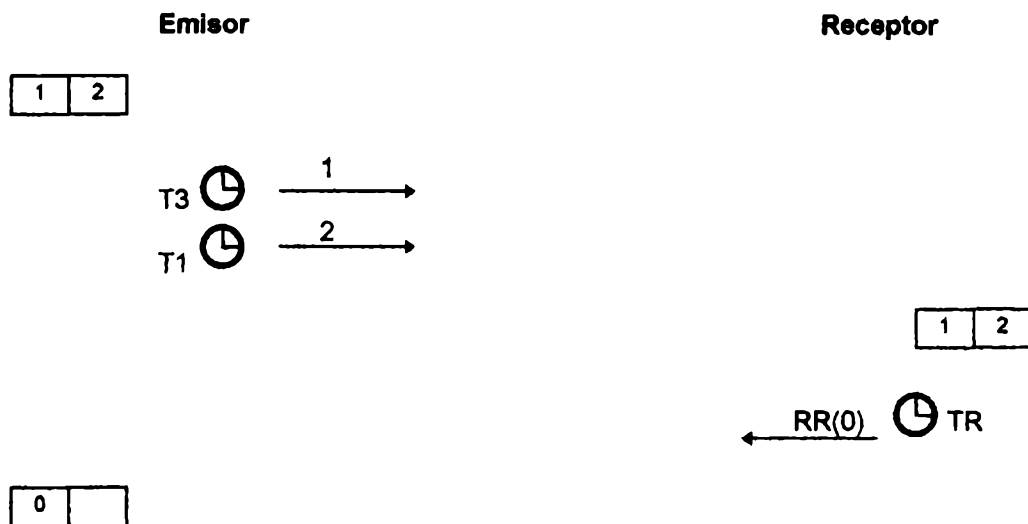
- **evento** : consiste en la activación y la desactivación de las transiciones
- **paso**: cubre la desactivación de todas las transiciones en el mismo tiempo (sin las nuevas activadas) y las activaciones resultantes de las transiciones
- **tiempo**: tiempo del proceso desde el comienzo de la simulación
- **transición**: nombre de la transición
- **estado de la transición**: puede estar en on: activada o en off: desactivada.

A.1 Ejemplo 1: Transmisión Normal

evento	paso	tiempo	transic.	estado	
337	122	118	t2	off	
338	122	118	t3	on	
339	122	118	t10	on	arma t1
340	123	120	t3	off	
341	123	120	t11	on	arma t2
342	124	121	t10	off	
343	124	121	t88	on	envía t1
344	124	121	t180	on	
345	125	121	t180	off	
346	125	121	t187	on	
347	126	121	t187	off	
348	126	121	tim3	on	activa T3 para t1
349	127	123	t11	off	

350	127	123	t88	off	
351	127	123	t180	on	
352	127	123	t62	on	llega t1
353	127	123	t90	on	envía t2
354	128	123	t180	off	
355	128	123	t106	on	
356	129	123	t106	off	
357	129	123	tim1	on	T1 para t2
358	130	125	t62	off	
359	130	125	t90	off	
360	130	125	t14	on	
361	130	125	t70	on	
362	130	125	t63	on	llega t2
363	130	125	t140	on	
364	131	125	t14	off	
365	131	125	t70	off	
366	131	125	t140	off	
367	131	125	t73	on	activa TR
368	131	125	t66	on	
369	132	125	t66	off	
370	133	126	t63	off	
371	133	126	t15	on	
372	133	126	t71	on	
373	133	126	t142	on	
374	134	126	t15	off	
375	134	126	t71	off	
376	134	126	t142	off	
377	134	126	t67	on	
378	134	126	t93	on	
379	135	126	t67	off	
380	135	126	t93	off	
381	136	129	tim2	off	
382	136	129	t111	on	
383	137	129	t111	off	
384	138	131	t73	off	vence TR
385	138	131	t30	on	confirma t1 y t2
386	139	132	t30	off	

387	139	132	t129	on	
388	139	132	t80	on	envía conf. t1 y t2
389	140	132	t129	off	
390	141	134	t80	off	
391	141	134	t204	on	llega confirmación
392	141	134	t236	on	
393	142	134	t204	off	
394	142	134	t236	off	
395	142	134	t161	on	
396	142	134	t1	on	
397	142	134	t156	on	
398	142	134	t212	on	
399	143	134	t161	off	
400	143	134	t156	off	
401	143	134	t212	off	



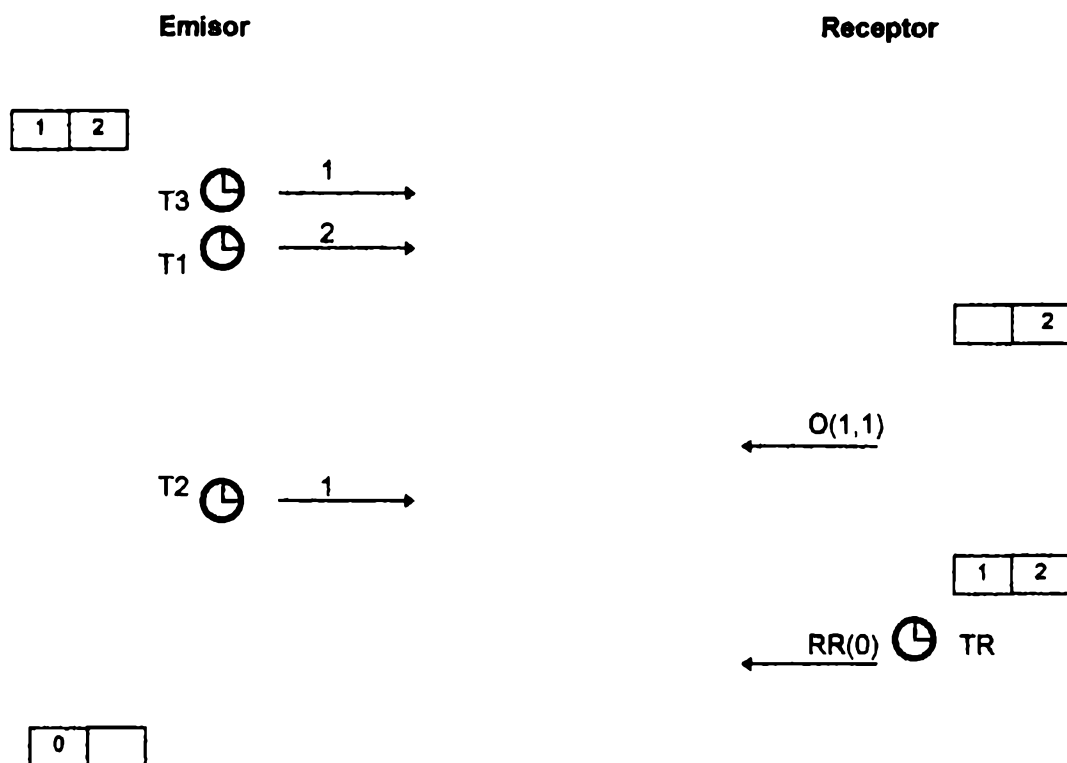
Aquí la transmisión de tramas sucede con total normalidad, como así los temporizadores se encuentran bien configurados. Se envían las tramas 1 y 2 arrancando los timers 3 y 1 respectivamente. Luego del arribo se pone en marcha el de recepción y con su vencimiento se confirman ambas. El asentimiento se presenta sin problemas al emisor que modifica su ventana.

A.2 Ejemplo 2.1: Pérdida de una trama de información (Ventana 2)

evento	paso	tiempo	transic.	estado
1881	650	559	t212	off
1882	651	561	t2	off
1883	651	561	t3	on
1884	651	561	t10	on arma t1
1885	652	563	t3	off
1886	652	563	t11	on arma t2
1887	653	564	tim1	off vence T1
1888	653	564	t10	off
1889	653	564	t89	on envía t1, luego se pierde
1890	653	564	t180	on
1891	653	564	t110	on
1892	654	564	t89	off
1893	654	564	t180	off
1894	654	564	t110	off
1895	654	564	t187	on
1896	655	564	t187	off
1897	655	564	tim3	on arranca T3 para t1
1898	656	566	tim2	off vence T2
1899	656	566	t11	off
1900	656	566	t90	on envía t2
1901	656	566	t180	on
1902	656	566	t111	on
1903	657	566	t180	off
1904	657	566	t111	off
1905	657	566	t106	on
1906	658	566	t106	off
1907	658	566	tim1	on activa T1 para t2
1908	659	568	t90	off
1909	659	568	t63	on llega t2
1910	660	570	t63	off
1911	660	570	t39	on envía O(1,1)
1912	660	570	t142	on
1913	661	570	t142	off
1914	662	571	t39	off
1915	662	571	t148	on O(1,1) en línea
1916	663	573	t148	off

1917	663	573	t215	on	llega O(1,1)
1918	664	573	t215	off	
1919	664	573	t157	on	
1920	664	573	t232	on	
1921	664	573	t10	on	arma t1
1922	665	573	t157	off	
1923	665	573	t232	off	
1924	666	576	t10	off	
1925	666	576	t88	on	envía t1
1926	666	576	t180	on	
1927	667	576	t180	off	
1928	667	576	t107	on	
1929	668	576	t107	off	
1930	668	576	tim2	on	activa T2(1)
1931	669	578	t88	off	
1932	669	578	t62	on	llega t1
1933	670	580	t62	off	
1934	670	580	t139	on	
1935	670	580	t14	on	
1936	670	580	t70	on	
1937	671	580	t139	off	
1938	671	580	t14	off	
1939	671	580	t70	off	
1940	671	580	t73	on	activa TR
1941	671	580	t66	on	
1942	671	580	t15	on	
1943	672	580	t66	off	
1944	672	580	t15	off	
1945	672	580	t67	on	
1946	672	580	t93	on	
1947	673	580	t67	off	
1948	673	580	t93	off	
1949	674	581	tim3	off	vence T3
1950	674	581	t190	on	
1951	675	581	t190	off	
1952	676	583	tim1	off	vence T1
1953	676	583	t110	on	

1954	677	583	t110	off	
1955	678	586	t73	off	vence TR
1956	678	586	t30	on	confirma t1 y t2
1957	679	587	t30	off	
1958	679	587	t129	on	
1959	679	587	t80	on	confirmación t1 y t2 en línea
1960	680	587	t129	off	
1961	681	589	t80	off	
1962	681	589	t204	on	llega confirmación
1963	681	589	t236	on	
1964	682	589	t204	off	
1965	682	589	t236	off	
1966	682	589	t161	on	
1967	682	589	t156	on	



Se envían las tramas 1 y 2, la primera se pierde y la segunda arriba correctamente. Los timeouts que suceden en estos intervalos no pertenecen a esta nueva secuencia, sino a la precedente. La trama 2 genera un pedido de retransmisión de la anterior. Con la llegada de la

solicitud al emisor se reemite la trama 1. la cual se presenta en el receptor, originando el asentimiento luego del vencimiento del temporizador de recepción.

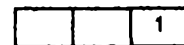
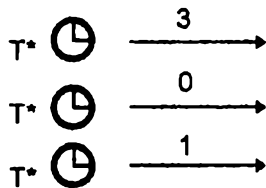
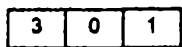
A.3 Ejemplo 2.2: Pérdida de dos Tramas de Información (Ventana 3)

evento	paso	tiempo	transic.	estado	
637	247	172	t4	on	
638	248	174	t4	off	
639	248	174	t8	on	envía t3 con pérdida posterior
640	248	174	t1	on	
641	249	174	t8	off	
642	250	176	t1	off	
643	250	176	t2	on	
644	250	176	t5	on	envía t0 con pérdida posterior
645	251	176	t5	off	
646	252	178	t2	off	
647	252	178	t10	on	envía t1
648	253	181	t10	off	
649	253	181	t62	on	llega t1
650	254	181	t62	off	
651	254	181	t44	on	envía O(3,0)
652	255	184	t44	off	
653	255	184	t56	on	
654	256	184	t56	off	
655	256	184	t12	on	envía t3
656	256	184	t60	on	
657	257	186	t60	off	
658	257	186	t9	on	envía t0
659	258	187	t12	off	
660	258	187	t64	on	
661	259	187	t64	off	
662	259	187	t16	on	
663	259	187	t72	on	
664	260	187	t16	off	
665	260	187	t72	off	
666	260	187	t73	on	arranca TR

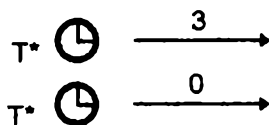
667	260	187	t68	on	
668	261	187	t68	off	
669	262	189	t9	off	
670	262	189	t61	on	
671	263	189	t61	off	
672	263	189	t13	on	
673	263	189	t69	on	
674	264	189	t13	off	
675	264	189	t69	off	
676	264	189	t65	on	
677	264	189	t17	on	
678	264	189	t14	on	
679	265	189	t65	off	
680	265	189	t17	off	
681	265	189	t14	off	
682	265	189	t66	on	
683	265	189	t74	on	
684	266	189	t66	off	
685	266	189	t74	off	
686	267	193	t73	off	vence TR
687	267	193	t27	on	confirma t3, t0 y t1
688	268	193	t27	off	

Emisor

Receptor



$\xleftarrow{O(3,0)}$





(*) Se diseñó otra RdP con el fin de presentar el pedido de retransmisión para tramas consecutivas. Debido a ello, se obviaron los timers de emisión, aunque en la realidad son necesarios razón por la cual se los incorporó en la secuencia de gráficos. La idea es mostrar la posibilidad de solicitar dos tramas consecutivas, trabajando con cuatro tramas y en consecuencia una ventana de tamaño tres.

En este ejemplo, se transmiten las tramas 3, 0 y 1. Esta última arriba al receptor y al notar que no tiene las precedentes, origina una petición para las tramas anteriores. A posteriori, se retransmiten las tramas solicitadas, es decir la 3 y la 0. A su llegada y luego del timeout del receptor se asienten las tres tramas.

A.4 Ejemplo 3: Pérdida de Tramas de Información de la Ventana Completa

evento	paso	tiempo	transic.	estado	
1	1	0	t1	on	
2	2	2	t1	off	
3	2	2	t2	on	
4	2	2	t9	on	arma t0
5	3	4	t2	off	
6	3	4	t10	on	arma t1
7	4	5	t9	off	
8	4	5	t87	on	puesta en línea de t0 y pérdida posterior
9	4	5	t180	on	
10	5	5	t87	off	
11	5	5	t180	off	
12	5	5	t187	on	
13	6	5	t187	off	
14	6	5	tim3	on	activa T3 para t0
15	7	7	t10	off	

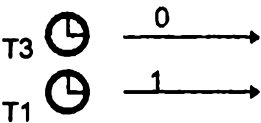
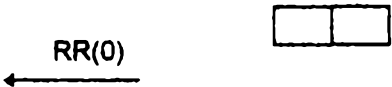
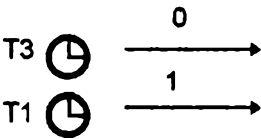
16	7	7	t180	on	
17	7	7	t89	on	puesta en línea de t1 y pérdida posterior
18	8	7	t180	off	
19	8	7	t89	off	
20	8	7	t106	on	
21	9	7	t106	off	
22	9	7	tim1	on	activa T1 para t1
23	10	22	tim3	off	vence T3
24	10	22	t190	on	
25	11	22	t190	off	
26	12	24	tim1	off	vence T1
27	12	24	time1	on	
28	13	24	time1	off	
29	13	24	t115	on	
30	14	24	t115	off	
31	14	24	t158	on	envía REQ: t0 y t1 pendientes
32	15	26	t158	off	
33	15	26	t177	on	se dispone a arrancar timer
34	15	26	t241	on	
35	15	26	t172	on	activa retransmisión de t0 y t1
36	16	26	t177	off	
37	16	26	t241	off	
38	16	26	t180	on	
39	17	26	t180	off	
40	17	26	t107	on	
41	18	26	t107	off	
42	18	26	tim2	on	activa T2 para el REQ
43	19	27	t172	off	
44	19	27	t224	on	
45	19	27	t152	on	
46	19	27	t234	on	
47	19	27	t9	on	arma t0
48	20	27	t152	off	
49	20	27	t234	off	
50	21	28	t224	off	
51	21	28	t157	on	
52	21	28	t10	on	arma t1

53	22	28	t157	off	
54	23	30	t9	off	
55	23	30	t180	on	
56	23	30	t86	on	t0 en línea
57	24	30	t180	off	
58	24	30	t187	on	
59	25	30	t187	off	
60	25	30	tim3	on	activa T3 para t0
61	26	31	t10	off	
62	26	31	t180	on	
63	26	31	t88	on	t1 en línea
64	27	31	t180	off	
65	27	31	t106	on	
66	28	31	t106	off	
67	28	31	tim1	on	activa T1 para t1
68	29	32	t86	off	
69	29	32	t61	on	llega t0 al receptor
70	30	33	t88	off	
71	30	33	t61	off	
72	30	33	t13	on	
73	30	33	t69	on	
74	30	33	t62	on	llega t1 al receptor
75	30	33	t138	on	
76	31	33	t13	off	
77	31	33	t69	off	
78	31	33	t138	off	
79	31	33	t73	on	activa TR
80	31	33	t65	on	
81	32	33	t65	off	
82	33	34	t62	off	
83	33	34	t14	on	
84	33	34	t70	on	
85	33	34	t140	on	
86	34	34	t14	off	
87	34	34	t70	off	
88	34	34	t140	off	
89	34	34	t66	on	


90	34	34	t93	on	
91	35	34	t66	off	
92	35	34	t93	off	
93	36	39	t73	off	vence TR
94	36	39	t26	on	confirma t0 y t1
95	37	40	t26	off	
96	37	40	t84	on	conf. t0 y t1 en línea
97	37	40	t125	on	
98	38	40	t125	off	
99	39	42	t84	off	
100	39	42	t200	on	llega conf. t0 y t1
101	39	42	t236	on	
102	40	42	t200	off	
103	40	42	t236	off	
104	40	42	t156	on	
105	40	42	t153	on	

Emisor

Receptor



0	1
---	---

← RR(2)  TR

2	
---	--

En este ej. se transmiten las tramas 0 y 1, poniéndose en marcha los temporizadores T3 y T1 respectivamente. Desgraciadamente, ambas tramas de información se pierden, generándose los timeouts y en consecuencia el disparo de un REQ. Ante la respuesta del receptor, el emisor reenvía las tramas 0 y 1, que en esta oportunidad llegan al destinatario. Luego, hay una puesta en marcha del temporizador de recepción de parte de la trama 0 con su consiguiente vencimiento. Se arma la confirmación correspondiente a las dos, y su posterior transmisión y llegada al emisor.

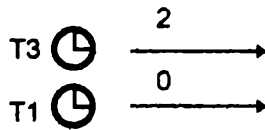
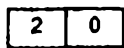
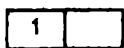
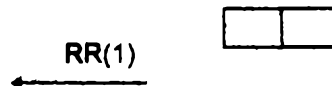
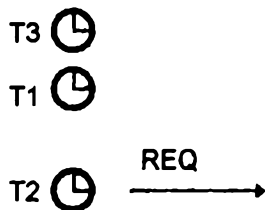
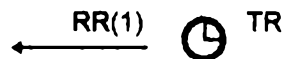
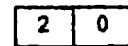
A.5 Ejemplo 4: Pérdida de la Confirmación

evento	paso	tiempo	transic.	estado	
238	89	86	t3	on	
239	89	86	t153	on	
240	89	86	t156	on	
241	89	86	t212	on	
242	90	86	t153	off	
243	90	86	t156	off	
244	90	86	t212	off	
245	91	88	t3	off	
246	91	88	t1	on	
247	91	88	t11	on	arma t2
248	92	90	t1	off	
249	92	90	t9	on	arma t0
250	93	91	tim2	off	vence T2
251	93	91	t11	off	
252	93	91	time2	on	
253	93	91	t90	on	t2 en línea
254	93	91	t180	on	
255	94	91	time2	off	
256	94	91	t180	off	

257	94	91	t187	on	
258	94	91	t115	on	
259	95	91	t187	off	
260	95	91	t115	off	
261	95	91	tim3	on	activa T3 para t2
262	95	91	t230	on	
263	96	91	t230	off	
264	97	93	t9	off	
265	97	93	t90	off	
266	97	93	t63	on	llega t2
267	97	93	t180	on	
268	97	93	t86	on	t0 en línea
269	98	93	t180	off	
270	98	93	t106	on	
271	99	93	t106	off	
272	99	93	tim1	on	activa T1 para t0
273	100	95	t63	off	
274	100	95	t86	off	
275	100	95	t61	on	llega t0
276	100	95	t15	on	
277	100	95	t71	on	
278	100	95	t142	on	
279	101	95	t15	off	
280	101	95	t71	off	
281	101	95	t142	off	
282	101	95	t73	on	activa TR
283	101	95	t67	on	
284	102	95	t67	off	
285	103	97	t61	off	
286	103	97	t13	on	
287	103	97	t69	on	
288	103	97	t138	on	
289	104	97	t13	off	
290	104	97	t69	off	
291	104	97	t138	off	
292	104	97	t65	on	
293	104	97	t93	on	

294	105	97	t65	off	
295	105	97	t93	off	
296	106	101	t73	off	vence TR
297	106	101	t34	on	confirma t2 y t0
298	107	102	t34	off	
299	107	102	t132	on	
300	107	102	t105	on	puesta en línea conf. t2 y t0 y posterior pérdida
301	108	102	t132	off	
302	108	102	t105	off	
303	109	108	tim3	off	vence T3
304	109	108	t190	on	
305	110	108	t190	off	
306	111	110	tim1	off	vence T1
307	111	110	time1	on	
308	112	110	time1	off	
309	112	110	t115	on	
310	113	110	t115	off	
311	113	110	t164	on	REQ: t2 y t0 pendientes
312	114	112	t164	off	
313	114	112	t177	on	
314	114	112	t241	on	
315	114	112	t121	on	
316	115	112	t177	off	
317	115	112	t241	off	
318	115	112	t180	on	
319	116	112	t180	off	
320	116	112	t107	on	
321	117	112	t107	off	
322	117	112	tim2	on	activa T2 para REQ
323	118	114	t121	off	
324	118	114	t76	on	conf. t2 y t0 en línea
325	119	116	t76	off	
326	119	116	t235	on	
327	119	116	t208	on	llega conf. t2 y t0
328	120	116	t235	off	
329	120	116	t208	off	
330	120	116	t2	on	

331	120	116	t153	on
332	120	116	t161	on
333	120	116	t212	on
334	121	116	t153	off
335	121	116	t161	off
336	121	116	t212	off

Emisor**Receptor**

El problema que surge en este ejemplo, si bien las tramas 2 y 0 llegan correctamente al receptor, es la pérdida del asentimiento. Con ello vencen los timers de emisión T3 y T1, determinando el envío de un REQ. Cuando dicha trama se pone en línea activa T2. A la llegada en el receptor, confirma las tramas 2 y 0.

A.6 Ejemplo 5: Miscelánea I

evento	paso	tiempo	transic.	estado	
106	40	42	t3	on	
107	40	42	t212	on	
108	41	42	t156	off	
109	41	42	t153	off	
110	41	42	t212	off	
111	42	43	tim2	off	vence T2
112	42	43	t111	on	
113	43	43	t111	off	
114	44	44	t3	off	
115	44	44	t1	on	
116	44	44	t11	on	arma t2
117	45	46	t1	off	
118	45	46	t9	on	arma t0
119	46	47	tim3	off	vence T3
120	46	47	t11	off	
121	46	47	t180	on	
122	46	47	t90	on	t2 en línea
123	46	47	t190	on	
124	47	47	t180	off	
125	47	47	t190	off	
126	47	47	t107	on	
127	48	47	t107	off	
128	48	47	tim2	on	activa T2 para t2
129	49	48	tim1	off	vence T1
130	49	48	t110	on	
131	50	48	t110	off	
132	51	49	t9	off	
133	51	49	t90	off	
134	51	49	t180	on	
135	51	49	t87	on	puesta en línea de t0 con posterior pérdida
136	51	49	t63	on	llega t2
137	52	49	t180	off	
138	52	49	t87	off	
139	52	49	t187	on	



140	53	49	t187	off	
141	53	49	tim3	on	activa T3 para t0
142	54	50	t63	off	
143	54	50	t15	on	
144	54	50	t71	on	
145	54	50	t142	on	
146	55	50	t15	off	
147	55	50	t71	off	
148	55	50	t142	off	
149	55	50	t73	on	activa TR
150	55	50	t67	on	
151	56	50	t67	off	
152	57	56	t73	off	vence TR
153	57	56	t32	on	confirma t2
154	58	57	t32	off	
155	58	57	t130	on	
156	58	57	t78	on	conf. t2 en línea
157	59	57	t130	off	
158	60	59	t78	off	
159	60	59	t206	on	llega conf. t2
160	60	59	t236	on	
161	61	59	t206	off	
162	61	59	t236	off	
163	61	59	t2	on	
164	61	59	t181	on	
165	61	59	t212	on	
166	62	59	t181	off	
167	62	59	t212	off	
168	63	61	t2	off	
169	63	61	t10	on	arma t1
170	64	64	tim2	off	vence T2
171	64	64	t10	off	
172	64	64	t180	on	
173	64	64	t88	on	t1 en línea
174	64	64	t111	on	
175	65	64	t180	off	
176	65	64	t111	off	

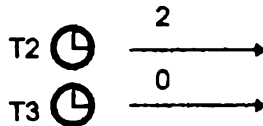
177	65	64	t106	on	
178	66	64	t106	off	
179	66	64	tim1	on	activa T1 para t1
180	67	66	tim3	off	vence T3
181	67	66	t88	off	
182	67	66	t62	on	llega t1
183	67	66	t190	on	
184	68	66	t190	off	
185	69	68	t62	off	
186	69	68	t37	on	envía O(0,0)
187	69	68	t140	on	
188	70	68	t140	off	
189	71	69	t37	off	
190	71	69	t147	on	O(0,0) en línea
191	72	71	t147	off	
192	72	71	t210	on	llega O(0,0)
193	73	71	t210	off	
194	73	71	t152	on	
195	73	71	t232	on	
196	73	71	t9	on	arma t0
197	74	71	t152	off	
198	74	71	t232	off	
199	75	74	t9	off	
200	75	74	t180	on	
201	75	74	t86	on	t0 en línea
202	76	74	t180	off	
203	76	74	t107	on	
204	77	74	t107	off	
205	77	74	tim2	on	activa T2 para t0
206	78	76	t86	off	
207	78	76	t61	on	llega t0
208	79	77	t61	off	
209	79	77	t137	on	
210	79	77	t13	on	
211	79	77	t69	on	
212	80	77	t137	off	
213	80	77	t13	off	

214	80	77	t69	off	
215	80	77	t73	on	activa TR
216	80	77	t14	on	
217	80	77	t65	on	
218	81	77	t14	off	
219	81	77	t65	off	
220	81	77	t66	on	
221	81	77	t93	on	
222	82	77	t66	off	
223	82	77	t93	off	
224	83	81	tim1	off	vence T1
225	83	81	t110	on	
226	84	81	t110	off	
227	85	83	t73	off	vence TR
228	85	83	t26	on	confirma t0 y t1
229	86	84	t26	off	
230	86	84	t125	on	
231	86	84	t84	on	conf. t0 y t1 en línea
232	87	84	t125	off	
233	88	86	t84	off	
234	88	86	t200	on	llega conf. t0 y t1
235	88	86	t236	on	
236	89	86	t200	off	
237	89	86	t236	off	

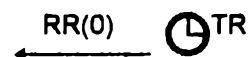
Emisor

Receptor

2	0
---	---

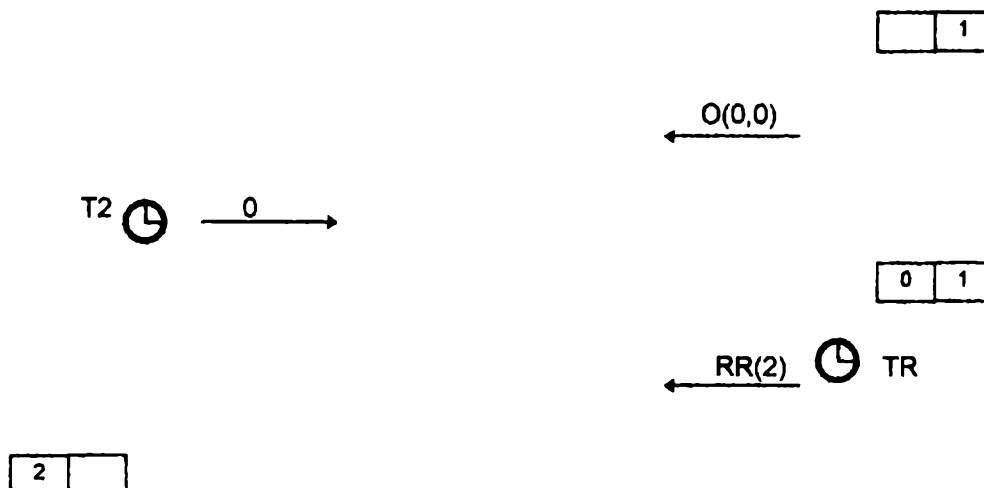


2	
---	--



0	1
---	---





Aquí hay una transmisión de la trama 2 que llega intacta al receptor, pero la que se pierde es su siguiente, la 0. Cuando arriba la 2 activa el temporizador, espera el timeout y después emite una confirmación para ella sola. El asentimiento arriba al emisor permitiendo correr la ventana y enviar la trama 1. Luego de despachada llega a destino y al notar la ausencia de su precedente realiza el pedido de retransmisión. Esta trama al arribar al emisor le exige la trama 0, que luego le es remitida. A continuación, se presenta la trama anterior al receptor, realizándose las gestiones atinentes para la confirmación de las dos tramas, es decir la 0 y la 1. La mecánica del asentimiento es similar a los anteriores.

A.7 Ejemplo 6: Miscelánea II

evento	paso	tiempo	translc.	estado	
3528	1209	1030	t1	off	
3529	1209	1030	t2	on	
3530	1209	1030	t9	on	arma t0
3531	1210	1032	tim3	off	
3532	1210	1032	t2	off	
3533	1210	1032	t190	on	
3534	1210	1032	t10	on	arma t1
3535	1211	1032	t190	off	
3536	1212	1033	t9	off	
3537	1212	1033	t180	on	
3538	1212	1033	t86	on	t0 en línea

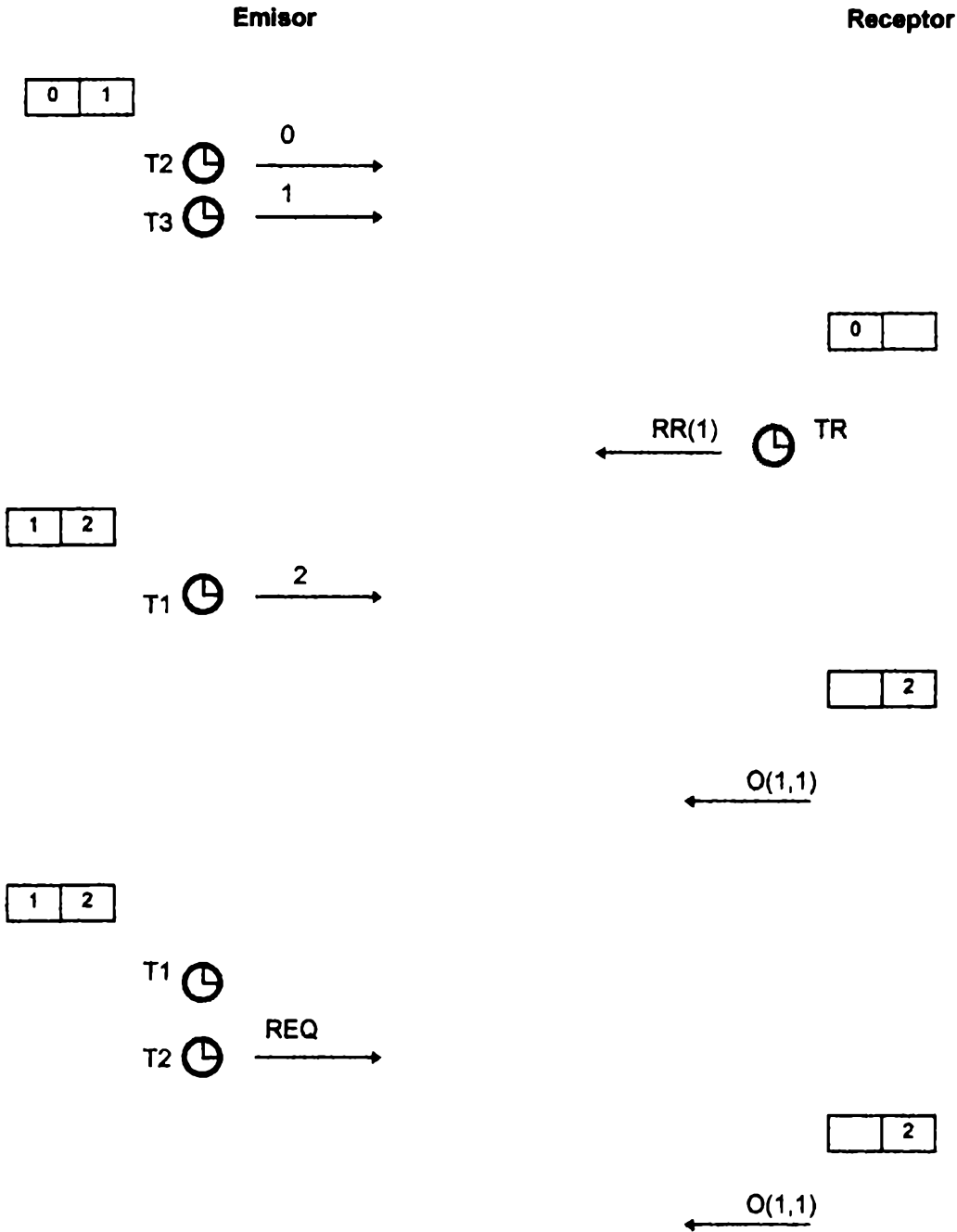
3539	1213	1033	t180	off	
3540	1213	1033	t107	on	
3541	1214	1033	t107	off	
3542	1214	1033	tim2	on	activa T2 para t0
3543	1215	1034	tim1	off	vence T1
3544	1215	1034	t110	on	
3545	1216	1034	t110	off	
3546	1217	1035	t10	off	
3547	1217	1035	t86	off	
3548	1217	1035	t61	on	llega t0
3549	1217	1035	t89	on	t1 en línea y posterior pérdida
3550	1217	1035	t180	on	
3551	1218	1035	t89	off	
3552	1218	1035	t180	off	
3553	1218	1035	t187	on	
3554	1219	1035	t187	off	
3555	1219	1035	tim3	on	activa T3 para t1
3556	1220	1036	t61	off	
3557	1220	1036	t13	on	
3558	1220	1036	t69	on	
3559	1220	1036	t138	on	
3560	1221	1036	t13	off	
3561	1221	1036	t69	off	
3562	1221	1036	t138	off	
3563	1221	1036	t73	on	activa TR
3564	1221	1036	t65	on	
3565	1222	1036	t65	off	
3566	1223	1042	t73	off	vence TR
3567	1223	1042	t25	on	confirma t0
3568	1224	1043	t25	off	
3569	1224	1043	t85	on	conf. t0 en línea
3570	1224	1043	t122	on	
3571	1225	1043	t122	off	
3572	1226	1045	t85	off	
3573	1226	1045	t198	on	llega conf. t0
3574	1226	1045	t236	on	
3575	1227	1045	t198	off	

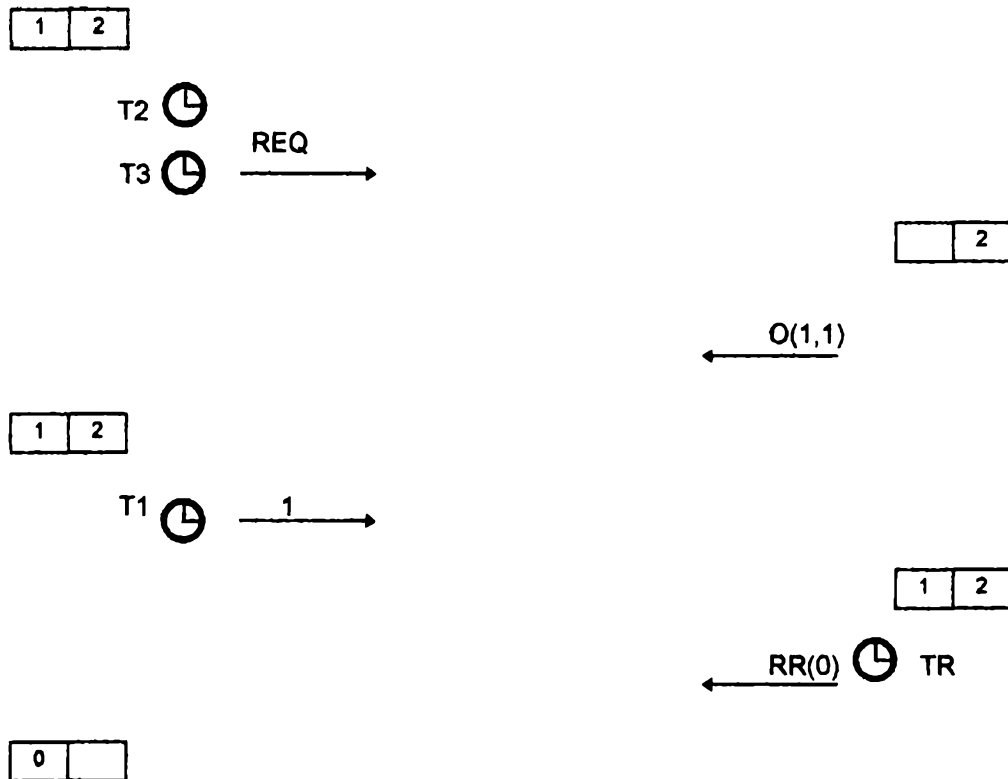
3576	1227	1045	t236	off	
3577	1227	1045	t3	on	
3578	1227	1045	t183	on	
3579	1227	1045	t212	on	
3580	1228	1045	t183	off	
3581	1228	1045	t212	off	
3582	1229	1047	t3	off	
3583	1229	1047	t11	on	
3584	1230	1050	tim2	off	vence T2
3585	1230	1050	t11	off	
3586	1230	1050	t90	on	t2 en línea
3587	1230	1050	t180	on	
3588	1230	1050	t111	on	
3589	1231	1050	t180	off	
3590	1231	1050	t111	off	
3591	1231	1050	t106	on	
3592	1232	1050	t106	off	
3593	1232	1050	tim1	on	T1 para t2
3594	1233	1052	tim3	off	vence T3
3595	1233	1052	t90	off	
3596	1233	1052	t63	on	llega t2
3597	1233	1052	t190	on	
3598	1234	1052	t190	off	
3599	1235	1053	t63	off	
3600	1235	1053	t39	on	arma O(1,1)
3601	1235	1053	t142	on	
3602	1236	1053	t142	off	
3603	1237	1054	t39	off	
3604	1237	1054	t149	on	O(1,1) en línea y posterior pérdida
3605	1238	1054	t149	off	
3606	1239	1067	tim1	off	vence T1
3607	1239	1067	time1	on	
3608	1240	1067	time1	off	
3609	1240	1067	t115	on	
3610	1241	1067	t115	off	
3611	1241	1067	t160	on	envía REQ: t1 y t2 pendientes
3612	1242	1069	t160	off	

3613	1242	1069	t177	on	
3614	1242	1069	t185	on	llega REQ
3615	1242	1069	t241	on	
3616	1243	1069	t177	off	
3617	1243	1069	t241	off	
3618	1243	1069	t180	on	
3619	1244	1069	t180	off	
3620	1244	1069	t107	on	
3621	1245	1069	t107	off	
3622	1245	1069	tim2	on	activa T2 para REQ
3623	1246	1071	t185	off	
3624	1246	1071	t149	on	O(1,1) en línea y posterior pérdida
3625	1247	1071	t149	off	
3626	1248	1086	tim2	off	vence T2
3627	1248	1086	time2	on	
3628	1249	1086	time2	off	
3629	1249	1086	t115	on	
3630	1250	1086	t115	off	
3631	1250	1086	t160	on	envía nuevo REQ
3632	1251	1088	t160	off	
3633	1251	1088	t185	on	llega REQ
3634	1251	1088	t177	on	
3635	1251	1088	t242	on	
3636	1251	1088	t241	on	
3637	1252	1088	t177	off	
3638	1252	1088	t242	off	
3639	1252	1088	t241	off	
3640	1252	1088	t180	on	
3641	1253	1088	t180	off	
3642	1253	1088	t187	on	
3643	1254	1088	t187	off	
3644	1254	1088	tim3	on	activa T3 para REQ
3645	1255	1090	t185	off	
3646	1255	1090	t148	on	O(1,1) en línea
3647	1256	1092	t148	off	
3648	1256	1092	t215	on	llega O(1,1)
3649	1257	1092	t215	off	

3650	1257	1092	t157	on	
3651	1257	1092	t237	on	
3652	1257	1092	t10	on	arma t1
3653	1258	1092	t157	off	
3654	1258	1092	t237	off	
3655	1259	1095	t10	off	
3656	1259	1095	t180	on	
3657	1259	1095	t88	on	envía t1
3658	1260	1095	t180	off	
3659	1260	1095	t106	on	
3660	1261	1095	t106	off	
3661	1261	1095	tim1	on	activa T1 para t1
3662	1262	1097	t88	off	
3663	1262	1097	t62	on	llega t1
3664	1263	1098	t62	off	
3665	1263	1098	t139	on	
3666	1263	1098	t14	on	
3667	1263	1098	t70	on	
3668	1264	1098	t139	off	
3669	1264	1098	t14	off	
3670	1264	1098	t70	off	
3671	1264	1098	t73	on	activa TR
3672	1264	1098	t66	on	
3673	1264	1098	t15	on	
3674	1265	1098	t66	off	
3675	1265	1098	t15	off	
3676	1265	1098	t67	on	
3677	1265	1098	t93	on	
3678	1266	1098	t67	off	
3679	1266	1098	t93	off	
3680	1267	1104	t73	off	vence TR
3681	1267	1104	t30	on	confirma t1 y t2
3682	1268	1105	tim3	off	vence T3
3683	1268	1105	t30	off	
3684	1268	1105	t128	on	
3685	1268	1105	t80	on	conf. t1 y t2 en línea
3686	1268	1105	t190	on	

3687	1269	1105	t128	off	
3688	1269	1105	t190	off	
3689	1270	1107	t80	off	
3690	1270	1107	t204	on	llega conf. t1 y t2
3691	1270	1107	t236	on	
3692	1271	1107	t204	off	
3693	1271	1107	t236	off	





En este caso se transmiten las tramas 0 y 1, de las cuales la segunda se pierde. Por lo tanto, confirma la 0, produciendo que el emisor envíe la trama 2. Cuando la anterior arriba al receptor, hace el pedido correspondiente a la trama 1, el cual no es reconocido en el destino. Al emisor le vence el timer y transmite REQ, al cual se responde con otra solicitud de la misma trama. Hay un nuevo extravío de ella, lo que provoca una nueva remisión del REQ, frente al timeout del emisor. El receptor contesta con el pedido que arriba al emisor. Se transmite la trama requerida, es decir la 1 originándose el asentimiento.

A.8 Ejemplo 7: Solapamiento de Tiempos

evento paso tiempo translc. estado

1	1	0	t1	on
2	2	2	t1	off
3	2	2	t2	on
4	2	2	t9	on
5	3	4	t2	off
6	3	4	t10	on

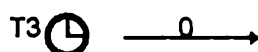
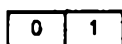
7	4	5	t9	off	
8	4	5	t86	on	envía t0
9	4	5	t180	on	
10	5	5	t180	off	
11	5	5	t187	on	
12	6	5	t187	off	
13	6	5	tim3	on	activa T3 para t0
14	7	7	t10	off	
15	7	7	t86	off	
16	7	7	t61	on	llega t0
17	7	7	t88	on	envía t1
18	7	7	t180	on	
19	8	7	t180	off	
20	8	7	t106	on	
21	9	7	t106	off	
22	9	7	tim1	on	activa T1 para t1
23	10	8	t61	off	
24	10	8	t13	on	
25	10	8	t69	on	
26	10	8	t138	on	
27	11	8	t13	off	
28	11	8	t69	off	
29	11	8	t138	off	
30	11	8	t73	on	activa TR
31	11	8	t65	on	
32	12	8	t65	off	
33	13	9	t88	off	
34	13	9	t62	on	llega t1
35	14	10	t62	off	
36	14	10	t14	on	
37	14	10	t70	on	
38	14	10	t140	on	
39	15	10	t14	off	
40	15	10	t70	off	
41	15	10	t140	off	
42	15	10	t66	on	
43	15	10	t93	on	

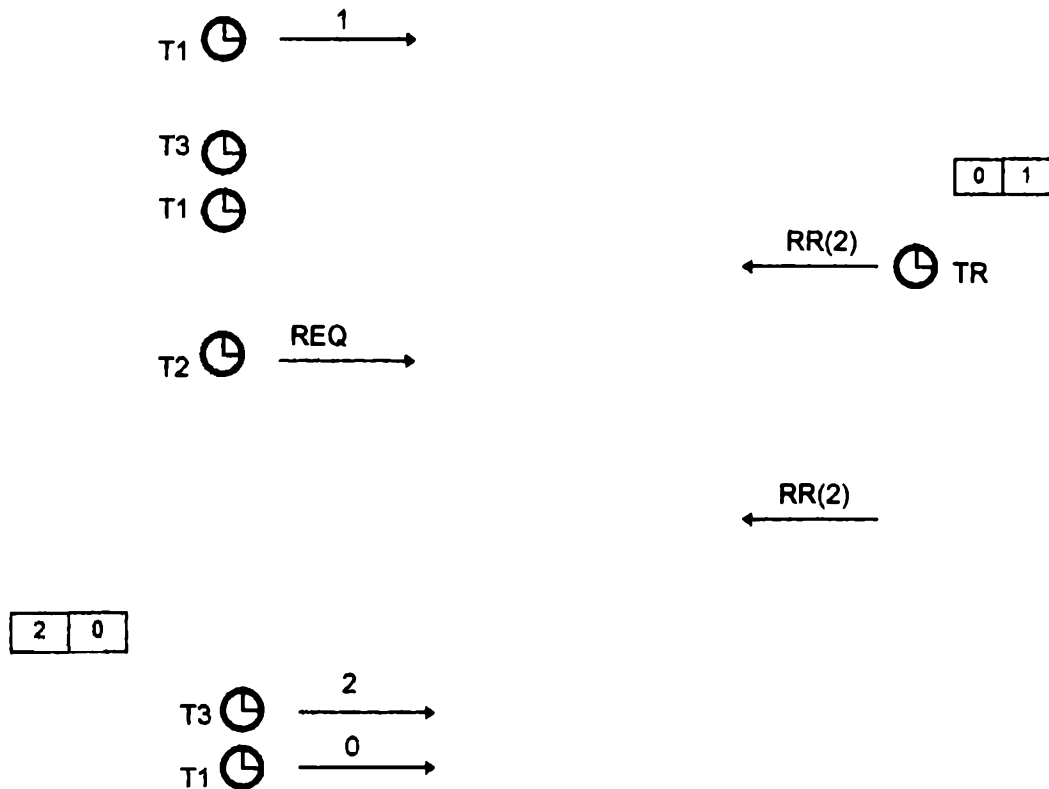
44	16	10	t66	off	
45	16	10	t93	off	
46	17	14	tim3	off	vence T3
47	17	14	t73	off	vence TR
48	17	14	t26	on	confirma t0 y t1
49	17	14	t190	on	
50	18	14	t190	off	
51	19	15	t26	off	
52	19	15	t125	on	
53	19	15	t84	on	conf. t0 y t1 en línea
54	20	15	t125	off	
55	21	16	tim1	off	vence T1
56	21	16	time1	on	
57	22	16	time1	off	
58	22	16	t115	on	
59	23	16	t115	off	
60	23	16	t158	on	envía REQ
61	24	17	t84	off	
62	24	17	t200	on	llega conf. t0 y t1 en línea
63	24	17	t236	on	
64	25	17	t200	off	
65	25	17	t236	off	
66	25	17	t3	on	
67	25	17	t156	on	
68	25	17	t153	on	
69	25	17	t212	on	
70	26	17	t156	off	
71	26	17	t153	off	
72	26	17	t212	off	
73	27	18	t158	off	
74	27	18	t177	on	
75	27	18	t241	on	
76	27	18	t117	on	reitera confirmación t0 y t1
77	28	18	t177	off	
78	28	18	t241	off	
79	28	18	t180	on	
80	29	18	t180	off	

81	29	18	t107	on	
82	30	18	t107	off	
83	30	18	tim2	on	activa T2 para REQ
84	31	19	t3	off	
85	31	19	t1	on	
86	31	19	t11	on	
87	32	20	t117	off	
88	32	20	t84	on	conf. t0 y t1 en línea
89	33	21	t1	off	
90	33	21	t9	on	
91	34	22	t11	off	
92	34	22	t84	off	
93	34	22	t235	on	
94	34	22	t90	on	envía t2
95	34	22	t180	on	
96	34	22	t199	on	llega conf. t0 y t1 (no modifica ventana)
97	35	22	t235	off	
98	35	22	t180	off	
99	35	22	t199	off	
100	35	22	t187	on	
101	36	22	t187	off	
102	36	22	tim3	on	activa T3 para t2
103	37	24	t9	off	
104	37	24	t90	off	
105	37	24	t180	on	
106	37	24	t86	on	envía t0
107	37	24	t63	on	
108	38	24	t180	off	
109	38	24	t106	on	
110	39	24	t106	off	
111	39	24	tim1	on	activa T1 para t0
112	40	26	t86	off	
113	40	26	t63	off	

Emisor

Receptor





Una mala configuración de los timers puede traer aparejado una mala utilización de la línea, produciendo retransmisiones innecesarias. En este caso hay una duplicación de confirmaciones, la segunda impulsada por el REQ. Si bien es un reenvío inútil, el protocolo resuelve el problema inhibiendo el segundo asentimiento. Se transmiten las tramas 0 y 1, las cuales se confirman. Cuando el asentimiento se encuentra en viaje, vence el último timer originando un REQ. En el emisor ante la llegada de la confirmación, corre la ventana y se dispone a emitir las nuevas tramas 2 y 0. En el receptor, ante la llegada del REQ envía nuevamente otro asentimiento para las tramas 0 y 1. Pese a que la confirmación arriba al emisor, no produce una acción determinada.

A.9 Ejemplo 8: Mala Configuración de Timers

evento	paso	tiempo	translc.	estado
1	1	0	t1	on
2	2	2	t1	off
3	2	2	t2	on
4	2	2	t9	on

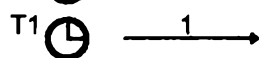
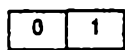
5	3	4	t2	off	
6	3	4	t10	on	
7	4	5	t9	off	
8	4	5	t180	on	
9	4	5	t86	on	envía t0
10	5	5	t180	off	
11	5	5	t187	on	
12	6	5	t187	off	
13	6	5	tim3	on	activa T3 para t0
14	7	7	t10	off	
15	7	7	t86	off	
16	7	7	t88	on	envía t1
17	7	7	t61	on	llega t0
18	7	7	t180	on	
19	8	7	t180	off	
20	8	7	t106	on	
21	9	7	t106	off	
22	9	7	tim1	on	activa T1 para t1
23	10	8	tim3	off	vence T3
24	10	8	t61	off	
25	10	8	t13	on	
26	10	8	t190	on	
27	11	8	t13	off	
28	11	8	t190	off	
29	11	8	t73	on	activa TR
30	12	9	t88	off	
31	12	9	t62	on	llega t1
32	13	10	tim1	off	vence T1
33	13	10	t62	off	
34	13	10	time1	on	
35	13	10	t14	on	
36	14	10	time1	off	
37	14	10	t14	off	
38	14	10	t115	on	
39	14	10	t93	on	
40	15	10	t115	off	
41	15	10	t93	off	

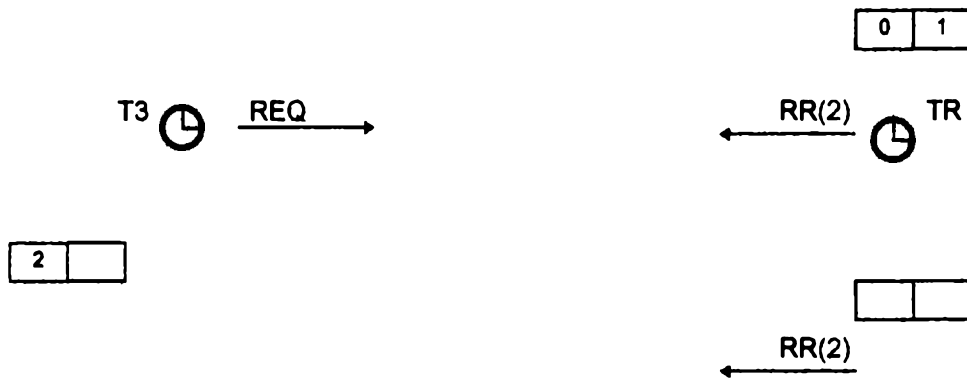
42	15	10	t158	on	envía REQ
43	16	12	t158	off	
44	16	12	t177	on	
45	16	12	t241	on	
46	16	12	t166	on	llega REQ, espera
47	17	12	t177	off	
48	17	12	t241	off	
49	17	12	t166	off	
50	17	12	t180	on	
51	18	12	t180	off	
52	18	12	t107	on	
53	19	12	t107	off	
54	19	12	tim2	on	activa T2 para REQ
55	20	14	t73	off	vence TR
56	20	14	t26	on	confirma t0 y t1
57	21	15	tim2	off	vence T2
58	21	15	t26	off	
59	21	15	time2	on	
60	21	15	t84	on	envía conf. t0 y t1
61	21	15	t125	on	
62	22	15	time2	off	
63	22	15	t125	off	
64	22	15	t115	on	
65	23	15	t115	off	
66	23	15	t158	on	envía REQ
67	24	17	t84	off	
68	24	17	t158	off	
69	24	17	t235	on	
70	24	17	t177	on	
71	24	17	t200	on	llega conf. t0 y t1
72	24	17	t241	on	
73	24	17	t117	on	activa nueva conf. t0 y t1
74	25	17	t235	off	
75	25	17	t177	off	
76	25	17	t200	off	
77	25	17	t241	off	
78	25	17	t3	on	

79	25	17	t180	on	
80	25	17	t153	on	
81	25	17	t156	on	
82	26	17	t180	off	
83	26	17	t153	off	
84	26	17	t156	off	
85	26	17	t187	on	
86	27	17	t187	off	
87	27	17	tim3	on	activa T3 para REQ
88	28	19	t117	off	
89	28	19	t3	off	
90	28	19	t1	on	
91	28	19	t84	on	envía conf. t0 y t1
92	28	19	t11	on	
93	29	20	tim3	off	vence T3
94	29	20	time3	on	
95	30	20	time3	off	
96	30	20	t115	on	
97	31	20	t115	off	
98	31	20	t230	on	
99	32	20	t230	off	
100	33	21	t1	off	
101	33	21	t84	off	
102	33	21	t235	on	
103	33	21	t199	on	llega conf. t0 y t1 (no modif. ventana)

Emisor

Receptor





Este es otro ejemplo, de lo mal que puede ser utilizado el ancho de banda cuando los tiempos se establecen inconscientemente. Para el mismo se creó otra RdP, para demostrar su funcionamiento. Se transmiten las tramas 0 y 1. Cuando se están procesando llega un REQ, provocando que este postergue su respuesta. Luego del vencimiento del timer del receptor, confirma las tramas de información. En ese instante ocurre un nuevo timeout del emisor generando otro REQ. A la llegada del asentimiento, el emisor corre la ventana disponiéndose a enviar nuevas tramas. Cuando el REQ arriba al receptor, se origina otra confirmación para las mismas tramas. Sin embargo, cuando ésta se presenta al transmisor no será tomada en cuenta. Por una mala configuración de los timers, se envían tres tramas innecesarias: dos REQ y un RR.

Apéndice B

ALGORITMOS

B. Algoritmos

B.1 Propuesta I

type

```
link = punt
a_link = registro
    pre, pos : link
    item : mensaje
end
```

var

```
buffer de emisión,
buffer de recepción : a_link
```

/* el buffer de emisión y de recepción son listas doblemente enganchadas */

buffer

begin

```
conexión_establecida = false
conectar = false
```

Do True

Procedure ENVIA_TRAMA (mensajeS)

begin

case mensajeS of

Info:

Construir trama de información

```
/* Arma la trama de información
Address := Comando (Receptor)
R := 0
N(S) := V(S)
V(S) := módulo N(V(S) + 1)
N(R) := V(R)
SEQ := SQE
if V(S) = 0
then
    SQE := modulo 2(SQE + 1)
end_if
Calcular CRC
CRC := CRC calculado
*/
```

```
Contador de retransmisión ( N(S) ) := 0
Guardar el paquete en el buffer de
emisión en el lugar correspondiente
if módulo(N(R) - 1) pertenece ventana recepción
then
```

/* hay tramas recibidas para confirmar
*/

```
    Enviar paquetes del buffer de recepción
    hasta modulo N ( N(R) - 1 ) a la Capa
    de Red
    Modificar Ventana de Recepción a
```


N(R)
end_if
 TimerE (N(S)) := on

retrans (J):
 Construir trama de información

If módulo(N(R) -1) pertenece ventana recepción
then

Enviar paquetes del buffer de recepción
 hasta modulo N (N(R) - 1) a la Capa
 de Red
 Modificar Ventana de Recepción a
 N(R)

end_if
 TimerE (N(S)) := on

RR :
 Construir trama RR

If módulo(N(R) -1) pertenece ventana recepción
then

Enviar paquetes del buffer de recepción
 hasta modulo N (N(R) - 1) a la Capa
 de Red
 Modificar Ventana de Recepción a
 N(R)

end_if

O :
 V(AR) := N(S)
 Construir trama O

Indicar pedido en el arreglo

/ Activa el timer correspondiente a la trama */*

/ Arma la trama a retransmitir:*
 Address := Comando (Receptor)
 R := 0
 N(S) := V(S)
 V(S) := módulo N(V(S) + 1)
 N(R) := V(R)
 SEQ := SQE
 if V(S) = 0
 then
 SQE := modulo 2(SQE + 1)
end_if
 Calcular CRC
 CRC := CRC calculado
**/*

/ hay tramas recibidas para confirmar */*

/ Activa el nuevamente el timer correspondiente a la trama retransmitida */*

/ Arma la trama RR*
 Address := Respuesta (Emisor)
 R := 0
 Código := 00
 SEQ := SQR
 N(R) := V(R)
 Calcular CRC
 CRC := CRC calculado
**/*

/ hay tramas recibidas para confirmar */*

/ Arma pedido de retransmisión:*
 Address := Comando (Emisor)
 R := 1
 N(S) := VAP(R)
 SEQ := SQE_VAP
 N(S+1) := módulo N(V(AR) - 1)
 Calcular CRC
 CRC := CRC calculado
*/**

/ for j := N(S) to N(S+1)*

ACC :

Construir trama ACC

Indicar aceptación en el arreglo

RNR :

Construir trama RNR

If módulo(N(R) -1) pertenece ventana recepción
then

Enviar paquetes del buffer de recepción
hasta modulo N (N(R) - 1) a la Capa
de Red

Modificar Ventana de Recepción a
N(R)

end_if

SABM :

Construir trama SABM

FRMR :

Construir trama FRMR

UA :

Construir trama UA

DM :

Construir trama DM

```
ARR(j) := 1
end_for */
```

```
/* Arma la trama de aceptación:
Address := Respuesta (Emisor)
R := 0
Código := 11
SEQ := SQR
N(R) := V(R)
Calcular CRC
CRC := CRC calculado
*/
/* ARR(V(R)) := 2 */
```

```
/* Arma la trama RNR:
Address := Respuesta (Emisor)
R := 0
Código := 00
SEQ := SQR
N(R) := V(R)
Calcular CRC
CRC := CRC calculado
*/
```

```
/* hay tramas recibidas para confirmar
*/
```

```
/* Arma trama SABM:
Address := Comando (Receptor)
R := 0
Código := 111100
Calcular CRC
CRC := CRC calculado
*/
```

```
/* Arma trama FRMR:
Address := Respuesta (Emisor)
R := 0
Código := 101001
Calcular CRC
CRC := CRC calculado
*/
```

```
/* Arma trama UA:
Address := Respuesta (Emisor)
R := 0
Código := 001110
Calcular CRC
CRC := CRC calculado
*/
```

```
/* Arma trama FRMR:
Address := Respuesta (Emisor)
R := 0
Código := 111000
Calcular CRC
CRC := CRC calculado
*/
```

end_case

SEND ch (mensajeS)

/ envía el mensaje */*

end */* Procedure ENVIA_TRAMA */*

Procedure RECIBE_TRAMA

begin

RECEIVE ch(mensajeR)

/ toma trama de la Capa Física */*

if no hay error de CRC

then

/ la trama no tiene errores de CRC */*

case mensajeR of

O :

if rango de secuencias pertenece
ventana

/ recibe pedido de retransmisión */*

/ busca en el buffer N(S) y N(S+1) y
que SEQ pertenezca a N(S) */*

then

/ N(S) y N(S+1) están en el buffer */*

if Recibí_RNR

then

/ habilito para poder enviar */*

Recibí_RNR = false

Seguir TimerR

/ sigue el timer de recepción */*

end_if

for j := N(S) to N(S + 1)

/ para todas las tramas del rango */*

TimerE (j) = off

/ desactiva el timer anterior */*

ENVIA_TRAMA (retrans(j))

/ retransmite la trama */*

end_for

else

/ el rango no está en la ventana */*

ENVIA_TRAMA (FRMR)

/ envía FRMR y pasa a la fase
de desconexión */*

Desconexión = true

end_if

RR:

/ recibe RR */*

if Recibí_RNR

then

/ habilito para poder enviar */*

Recibí_RNR := false

Seguir timerR

/ sigue el timer de recepción */*

end_if

if módulo N(N(R) - 1) pertenece a la ventana

/ confirma y modifica ventana */*

then

Correr la ventana hasta módulo N (N(R) - 1)

Desactivar todos los timers de las tramas

confirmadas

end_if

RNR:

/ recibe RNR */*

Deshabilita Capa de Red

if módulo N(N(R)-1) pertenece a la ventana

/ confirma y modifica ventana */*

then

Correr la ventana hasta módulo N(N(R) -1)

Desactiva los timers de las tramas confirmadas

Parar timerR

/ para el timer de recepción */*

Recibí_RNR = true

end_if

ACC:

if N(R) pertenece a la ventana

/ N(R) está en el buffer */*

then

TimerE (N(R)) := off

/ deshabilita el timer para esa trama */*

```

if Recibi_RNR
  then
    Recibi_RNR = true
    Seguir_TimerR
  end_if
end_if
INFO:
if longitud (campo de datos) > longitud máxima
  then
    ENVIA_TRAMA (FRMR)
    Desconexión = true
  else
if hay error en campo de datos
  then
    ENVIA_TRAMA (FRMR)
    Desconexión = true
  else
if trama pedida por retransmisión
  then
if N(S) viene en secuencia
  then
    Indicar la llegada en secuencia en el arreglo
    Guardar paquete en el buffer
    en el lugar correspondiente
    if módulo N(N(R) - 1) pertenece ventana emisión
    then
      Modificar ventana de emisión a N(R)
      Desactiva los timers de las tramas confirmadas
    end_if
if timerR = off
    then
      timerR := on
    end_if
if llegaron todas las tramas pedidas
    then
      Resetear el arreglo en 0
      Resetear variables de estado auxiliares
    end_if
  else
if N(S) cae en la ventana
  then
    Guardar el paquete en el buffer
    en el lugar correspondiente
    if módulo N(N(R) - 1) pertenece ventana emisión
    then
      Modificar ventana de emisión a N(R)
      Desactiva los timers de las tramas confirmadas
    end_if
if trama módulo N(N(S) - 1) llegó
    then
  
```

/* habilito para poder enviar */

/* sigue el timer de recepción */

/* recibe trama de información */

/* la longitud del campo de datos
sobrepasa a la máxima *//* envía FRMR y pasa a la fase
de desconexión */

/* el campo de datos es anormal */

/* envía FRMR y pasa a la fase
de desconexión */

/* trama pedida por retransmisión */

/* N(S) = V(R) y SEQ = SQR */

/* ARR(V(R)) := 3 */

/* confirma y modifica ventana */

/* el timer de recepción está
desactivado */

/* activarlo */

/* llegaron todas las tramas pedidas */

/* N(S) no viene en secuencia */

/* fuera de secuencia pero cae en la
ventana */

/* confirma y modifica ventana */

/* llegó la trama anterior */

/* envía ACC para N(S) */

```

        ENVIA_TRAMA (O)
        ENVIA_TRAMA (ACC)
    end_if
    TimerE(N(S)) = off
    else
        ENVIA_TRAMA (FRMR)
        Desconexión = true
    end_if
else
    If N(S) viene en secuencia
    then
        Guardar paquete en el buffer
        en el lugar correspondiente
        If módulo N(N(R) - 1) pertenece ventana emisión
        then
            Modificar ventana de emisión a N(R)
            Desactiva los timers de las tramas confirmadas
        end_if
        if timerR = off
        then
            timerR := on
        end_if
    else
        if N(S) cae en la ventana
        then
            Guardar paquete en el buffer
            en el lugar correspondiente
            if módulo N(N(R) - 1) pertenece ventana emisión
            then
                Modificar ventana de emisión a N(R)
                Desactiva los timers de las tramas confirmadas
            end_if
            If trama módulo N(N(S) - 1) llegó
            then
                ENVIA_TRAMA (ACC)
            else
                ENVIA_TRAMA (O)
                ENVIA_TRAMA (ACC)
            end_if
            TimerE (N(S)) = off
        else
            ENVIA_TRAMA (FRMR)
            Desconexión = true
        end_if
    end_if
end_if
otherwise
    ENVIA_TRAMA (FRMR)
    Desconexión = true
end_case
else

```

/ envía O */*
/ envía ACC para N(S) */*

/ desactiva el timer de esa trama */*
/ N(S) no cae en la ventana */*
/ envía FRMR y pasa a la fase de desconexión */*

/ trama de información que llega por primera vez */*
/ N(S) = V(R) y SEQ = SQR */*

/ confirma y modifica ventana */*

/ el timer recepción está desactivado */*
/ lo activa */*

/ no viene en secuencia */*

/ viene fuera de secuencia pero cae en la ventana */*

/ confirma y modifica ventana */*

/ llegó la trama anterior a N(S) */*
/ envía ACC para N(S) */*
/ no llegó la anterior */*
/ envía O */*
/ envía ACC para N(S) */*

/ viene fuera de secuencia y N(S) no cae en la ventana */*
/ envía FRMR y pasa a la fase de desconexión */*

/ trama no standard */*
/ envía FRMR y pasa a la fase de desconexión */*

/ error de CRC en trama */*

```

mensajeR = EMPTY
end

```

Procedure CONEXION

```

begin

```

```

    Timer = off

```

```

    i = 1

```

```

    repeat

```

```

        If (not EMPTY ch)

```

```

        then

```

```

            RECEIVE ch (mensajeR)

```

```

            case mensajeR of

```

```

                SABM : if (está en condiciones de conectarse)

```

```

                    then

```

```

                        ENVIA_TRAMA (UA)

```

```

                        conectar = true

```

```

                    else

```

```

                        ENVIA_TRAMA (DM)

```

```

                    end_if

```

```

                UA : If mensajeS = SABM

```

```

                    then

```

```

                        conexión_establecida = true

```

```

                    end_if

```

```

            end_case

```

```

        end_if

```

```

    if (quiere conectarse) and (not conexión_establecida)

```

```

        and (not conectar)

```

```

    then

```

```

        If (i<=n)

```

```

        then

```

```

            If timer = off

```

```

            then

```

```

                timer = on

```

```

                ENVIA_TRAMA (SABM)

```

```

                i = i+1

```

```

            end_if

```

```

        else

```

```

            Resuelve capas superiores

```

```

        end_if

```

```

    end_if

```

```

until (i>n) or (conexión_establecida) or (conectar)

```

```

if conectar

```

```

    then

```

```

        If (not EMPTY ch)

```

```

        then

```

```

            RECIBE_TRAMA

```

```

            If (mensajeR = SABM)

```

```

            then

```

```

                ENVIA_TRAMA (UA)

```

```

            else

```

```

                If mensajeR <> EMPTY)

```

```

/* Escucha el canal y verifica si existe
   pedido de conexión */

```

```

/* recibe SABM */

```

```

/* se conecta */

```

```

/* envía UA */

```

```

/* no se conecta */

```

```

/* envía DM */

```

```

/* recibe UA */

```

```

/* si antes envió SABM se establece la
   conexión */

```

```

/* se quiere conectar */

```

```

/* testea la cantidad de
   retransmisiones de SABM */

```

```

/* el timer está desactivado */

```

```

/* lo activa */

```

```

/* envía SABM */

```

```

/* retransmitió n veces SABM */

```

```

/* se contestó con UA un SABM */

```

```

/* escucha */

```

```

/* recibe SABM otra vez */

```

```

/* no recibe SABM */

```

```

                and (mensajeR <> DISC)
            then
                conexión_establecida = true
            end_if
        end_if
    end_if
end /* Procedure CONEXION */

```

/* el mensaje recibido no es DISC, la
conexión queda establecida */

Procedure TRANSFERENCIA

evento = (Recibiendo, Enviando, Ultimo, Timeout_R,
CapaRedOciosa, Timeout_E, Capa_Enlace_Ocupada,
Atención_Concedida, Enviado)

Procedure RECIBE

```

begin
    if (not EMPTY ch)
    then
        RECIBE_TRAMA
    end_if
end /* Procedure RECIBE */

```

/* escucha y recibe algo */

Procedure ENVIA

```

begin
    if (Capa de Red está habilitada)
        and (not Recibí_RNR)
        and (hay lugar en la ventana)
    then
        Obtiene nuevo paquete
        ENVIA_TRAMA (info)
    end_if
end /* Procedure ENVIA */

```

/* se puede enviar trama/s de info */

/* envía trama de información */

Procedure ULTIM

```

begin
    Desconexión = true
    ENVIA_TRAMA (DISC)
end /* Procedure ULTIM */

```

/* no hay más que hacer */

/* pasa a la fase de desconexión */

Procedure TIMEOUT_RECEPTOR

```

begin
    if TimerR = off
    then
        if (not Recibí_RNR)
        then
            if (hay tramas pendientes de confirmación)
            then
                ENVIA_TRAMA (RR)
            end_if
        end_if
    end_if
end /* Procedure TIMEOUT_RECEPTOR */

```

/* venció el timer de recepción */

/* está habilitado el envío */

/* queda/n trama/s pendiente/s de
confirmación */

/* envía RR */

```

end_if
end /* Procedure TIMEOUT_RECEPTOR */

```

```

Procedure CAPA_RED_OCIOsa

```

```

begin

```

```

  if (Capa Red está ociosa) and (not Recibí_RNR)
  then

```

```

/* el envío está habilitado y la Capa de
  Enlace está ociosa */

```

```

    ENVIA_TRAMA (RR)

```

```

/* envía RR */

```

```

  end_if

```

```

end /* Procedure CAPA_RED_OCIOsa */

```

```

Procedure TIMEOUT_EMITOR

```

```

begin

```

```

  if (Buffer_Emisión <> EMPTY)
  then

```

```

/* hay trama/s pendiente/s de
  confirmación */

```

```

    N(S) := Primer trama enviada

```

```

    Repeat

```

```

      If TimerE ( N(S) ) = off

```

```

/* venció el timer de la trama N(S) */

```

```

      then

```

```

        If (not Recibí_RNR)

```

```

/* el envío está habilitado */

```

```

        then

```

```

          If Contador de Retransm. ( N(S) ) <= K
          retransmisiones

```

```

/* testeo de la cantidad de retrans-
  misiones de la trama N(S) */

```

```

          then

```

```

            Contador de Retransm. ( N(S) ) :=

```

```

            Contador de Retrans. ( N(S) ) + 1

```

```

            ENVIA_TRAMA ( retrans ( N(S) )))

```

```

/* envía la retransmisión de la trama
  N(S) */

```

```

          else

```

```

/* hubo más de k retransmisiones para
  la trama N(S) */

```

```

            ENVIA_TRAMA (DISC)

```

```

            Desconexión = true

```

```

/* envía DISC y pasa a la fase de
  desconexión */

```

```

          end_if

```

```

        end_if

```

```

      end_if

```

```

    until (Desconexión) or (última trama enviada)
      or (Capa Enlace Ocupada)

```

```

end /* Procedure TIMEOUT_EMITOR */

```

```

begin

```

```

  Habilita_Capa_Red

```

```

  conectar := false

```

```

wait (evento)

```

```

/* espera un evento */

```

```

repeat

```

```

  case evento of

```

```

    Recibiendo      : Recibe

```

```

    Enviando        : Envía

```

```

    Ultimo          : Ultim

```

```

    TimeOut_R       : TimeOut_Receptor

```

```

    CapaRedOciosa   : Capa_Red_Ociosa

```

```

    TimeOut_E       : TimeOut_Emisor

```

```

  end_Case

```

```

  if evento = Capa_Enlace_Ocupada

```



```

then
    Envie_RNR = true
    ENVIA_TRAMA(RNR)
    Deshabilita_Capa_Red
end_if
until (ENVIA_RNR) or Desconexión
if not Desconexión
    then
        if evento = Atención_Concedida
            then
                Habilita_Capa_Red
                If (hay trama pendiente de aceptación)
                    then
                        ENVIA_TRAMA (ACC)
                    else
                        ENVIA_TRAMA (RR)
                    end_if
                Envie_RNR = false
            else
                If not EMPTY ch
                    then
                        RECEIVE ch (mensaje)
                        ENVIA_TRAMA (RNR)
                        mensajeR = EMPTY
                    end_if
                end_if
            end_if
        end
    /* Procedure TRANSFERENCIA */

```

Procedure DESCONEXION

```

if not EMPTY ch
    then
        RECEIVE (mensajeR)
        case mensajeS of
            FRMR : If (mensajeR = SABM) or (mensajeR = DISC)
                then
                    Final := true
                else
                    If (mensajeR <> DM)
                        then
                            ENVIA_TRAMA (FRMR)
                        end_if
                    DISC : if (mensajeR = SABM) or (mensajeR = UA)
                        then
                            Final := true
                            If (mensajeR = SABM)
                                then
                                    If está en condiciones de conectarse
                                        then
                                            ENVIA_TRAMA (UA)
                                            Conectar = true

```

/* la Capa de Enlace está ocupada */

/* envía RNR */

/* se envió RNR anteriormente */

/* se desocupa al Capa de Enlace */

/* hay trama pendiente de aceptación */

/* envía ACC */

/* no hay trama pendiente de aceptación */

/* envía RR */

/* realiza evento = Enviado, la Capa de Enlace sigue ocupada */

/* recibe algo */

/* envía nuevamente otro RNR */

/* recibe algo */

/* se envió FRMR */

/* se recibe SABM o DISC */

/* sale de la fase de desconexión */

/* se recibe algo que no es ni SABM, ni DISC ni DM */

/* envía nuevamente FRMR */

/* se envió DISC */

/* se recibe SABM o UA */

/* sale de la fase de desconexión */

/* pasa a la fase de conexión */

/* se conecta */

/* envía UA */

```

        else
            ENVIA_TRAMA (DM)
        end_if
    end_if
    else
        if mensajeR <> DM
            then
                ENVIA_TRAMA (DISC)
            end_if
        end_case
    else
        If i <= n
            then
                If Timer = off
                    then
                        Timer = on
                        ENVIA_TRAMA (mensajeS)
                        i := i + 1
                    end_if
                else
                    Final = true
                end_if
            end
        end /* Procedure DESCONEXION */

begin

final := false
while (not conexión_establecida) do
    CONEXIÓN
end_while

conexión_establecida := false
conectar := false
while (not desconexión) do
    TRANSFERENCIA
end_while

desconexión := false
while (not final) do
    DESCONEXIÓN
end_while

end

end_do

end.

```

```

/* no se conecta */
/* envía DM */

/* se recibe algo <> a SABM */

/* se recibe algo que no es ni SABM ni
DM */
/* envía DISC */

/* no recibió nada */
/* testea la cantidad de
retransmisiones de lo enviado */

/* venció el timer */
/* activa el timer */
/* retransmite la trama anterior */

/* las retransmisiones sobrepasaron
las n */
/* sale de la fase de desconexión */

```

B.2 Propuesta II

type

```
link = punt
a_link = registro
    pre, pos : link
    item : mensaje
end
```

var

```
buffer de emisión,
buffer de recepción : a_link
```

/ el buffer de emisión y de recepción son listas doblemente enganchadas */*

```
conexión_establecida = false
conectar = false
begin
Do True
```

Procedure ENVIA_TRAMA (mensajeS)

/ Procesa una trama para enviarla */*

begin

case mensajeS of

Info:

Construir trama de info

/ Arma la trama de información:
Address := Comando (Receptor)
R := 0
N(S) := V(S)
V(S) := módulo N(V(S) + 1)
P/F := P
N(R) := V(R)
Calcular CRC
CRC := CRC calculado
/

Guardar el paquete en el buffer de
emisión en el lugar correspondiente
if módulo N(N(R) - 1) pertenece
a ventana de recepción

then

/ hay tramas recibidas para confirmar
/**

Enviar paquetes del buffer de
recepción hasta modulo N (N(R) - 1)
a la Capa de Red
Modificar Ventana de Recepción a
N(R)

end_if

TimerE := on

/ Activa el timer de emisión */*

retrans (J):

Construir trama de información

/ Arma la trama a retransmitir:
Address := Comando (Receptor)
R := 0
N(S) := V(S)
V(S) := módulo N(V(S) + 1)
N(R) := V(R)
P/F := P
Calcular CRC
CRC := CRC calculado*

```

*/

if módulo  $N(N(R) - 1)$  pertenece
  a ventana de recepción
then
  Enviar paquetes del buffer de
  recepción hasta modulo  $N ( N(R) - 1 )$ 
  a la Capa de Red
  Modificar Ventana de Recepción a
   $N(R)$ 
end_if
TimerE := on
RR :

If no hay pedido de retransmisión
  enviado pendiente
then
  Construir trama RR

If módulo  $N(N(R) - 1)$  pertenece
  a ventana de recepción
then
  Enviar paquetes del buffer de
  recepción hasta modulo  $N ( N(R) - 1 )$ 
  a la Capa de Red
  Modificar Ventana de Recepción a
   $N(R)$ 
end_if
último := mensajes
end_if
O :
 $V(AR) := N(S)$ 
Construir trama O

Indicar pedido en el arreglo

último := mensajes
REQ :
  Construir trama REQ

/* hay tramas recibidas para confirmar
*/

/* activa el timer de emisión */
/* Arma la trama RR */
Address := Respuesta (Emisor)
R := 0
Código := 00
P/F := P/F
 $N(R) := V(R)$ 
Calcular CRC
CRC := CRC calculado */

/* testea porque prioriza un O a un RR
*/

/* Arma la trama RR */

/* hay trama/s recibida/s para
confirmar */

/* guarda en último el RR */

/* arma el pedido de retransmisión:
Address := Comando (Emisor)
R := 1
 $N(S) := VAP(R)$ 
P/F := P/F
 $N(S + 1) := \text{módulo } N(V(AR) - 1)$ 
Calcular CRC
CRC := CRC calculado */

/* for j :=  $N(S)$  to  $N(S + 1)$ 
  ARR(j) := 1
end_for */

/* guarda el RR enviado en último */

/* arma la trama REQ:
Address := Comando (Receptor)
R := 1
Código := 01
P/F := P
Resto := Resto
Calcular CRC
CRC := CRC calculado */

```

```

TimerE := on
RNR :
    Construir trama RNR

    If módulo  $N(N(R) - 1)$  pertenece
        a ventana de recepción
    then

        Enviar paquetes del buffer de
        recepción hasta modulo  $N ( N(R) - 1 )$ 
        a la Capa de Red
        Modificar Ventana de Recepción a
         $N(R)$ 
    end_if
SABM :
    Construir trama SABM

FRMR :
    Construir trama FRMR

UA :
    Construir trama UA

DM :
    Construir trama DM

end_case

SEND ch (mensajeS)

```

/* activa el timer de emisión */

/* arma la trama RNR:

```

Address := Respuesta (Emisor)
R := 0
Código := 10
P/F := P/F
N(R) := V(R)
Calcular CRC
CRC := CRC calculado */

```

/* hay trama/s recibida/s para confirmar */

/* arma trama SABM:

```

Address := Comando (Receptor)
R := 0
Código1 := 1111
P/F := P
Código2 := 100
Calcular CRC
CRC := CRC calculado */

```

/* arma trama FRMR:

```

Address := Respuesta (Emisor)
R := 0
Código1 := 1110
P/F := F
Código2 := 001
Calcular CRC
CRC := CRC calculado */

```

/* arma trama UA:

```

Address := Respuesta (Emisor)
R := 0
Código1 := 1100
P/F := F
Código2 := 110
Calcular CRC
CRC := CRC calculado */

```

/* arma trama DM:

```

Address := Respuesta (Emisor)
R := 0
Código1 := 1111
P/F := P
Código2 := 000
Calcular CRC
CRC := CRC calculado */

```

/* envía la trama */

end /* Procedure ENVIA_TRAMA */

Procedure RECIBE_TRAMA

begin

RECEIVE ch(mensajeR)

If no hay error de CRC

then

case mensajeR of

O :

If rango de secuencias

pertenece a ventana

then

If Recibí_RNR

then

Recibí_RNR = false

end_if

If ((VREQ = false) and (Envie_REQ))

or (not ENVIE_REQ)

then

for j := N(S) to N(S+1)

TimerE = off

ENVIA_TRAMA (retrans(j))

end_for

end_if

If Envie_REQ

then

VREQ = true

Envie_REQ = false

end_if

else

ENVIA_TRAMA (FRMR)

Desconexión = true

end_if

RR:

If Recibí_RNR

then

Recibí_RNR = false

end_if

If módulo N(N(R) - 1) pertenece a la ventana

then

Correr la ventana hasta módulo N (N(R) - 1)

If TimerE fue activado con

la última trama confirmada

then

TimerE := off

end_if

end_if

If Envie_REQ

then

If no hay tramas de información enviadas

/* toma trama de la Capa Física */

/* la trama no tiene errores de CRC */

/* recibe pedido de retransmisión */

/* busca en el buffer N(S) y N(S + 1) */

/* N(S) y N(S + 1) están en el buffer */

/* habilita para poder enviar */

/* enví REQ y no tengo respuesta o
está en condiciones normales */

/* para todas las tramas del rango */

/* desactiva el timer de emisión para
poder enviar nuevamente */

/* retransmite la trama */

/* el rango no está en la ventana */

/* envía FRMR y pasa a la fase
de desconexión */

/* recibe RR */

/* habilita el envío de tramas */

/* confirma y modifica ventana */

/* desactivar el timer de emisión
porque se confirmaron todas
las tramas */

/* se envió anteriormente REQ */


```

    la última trama confirmada
    then
        TimerE := off
        /* desactivar el timer de emisión
        porque se confirmaron todas
        las tramas */

    end_if
end_if
if timerR = off
    then
        /* el timer de recepción está
        desactivado */
        timerR := on
        /* activarlo */

    end_if
    if llegaron todas las tramas pedidas
        then
            /* llegaron todas las tramas pedidas */
            Resetea el arreglo en 0
            Resetea variables de estado auxiliares
        end_if
    else
        /* N(S) no viene en secuencia */
        if N(S) cae en la ventana
            then
                /* fuera de secuencia pero cae en la
                ventana */

                Guardar el paquete en el buffer
                en el lugar correspondiente
                if módulo  $N(N(R) - 1)$  pertenece
                ventana emisión
                    then
                        /* confirma y modifica ventana */
                        Modificar ventana de emisión a  $N(R) - 1$ 
                        if TimerE fue activado con
                        la última trama confirmada
                            then
                                /* desactivar el timer de emisión
                                porque se confirmaron todas
                                las tramas */
                                TimerE := off

                            end_if
                        end_if
                    if trama módulo  $N(N(S) - 1)$  no llegó
                    then
                        /* no llegó la trama anterior */
                        ENVIA_TRAMA (O)
                        /* envía O para tramas faltantes */
                    end_if
                else
                    /* N(S) no cae en la ventana */
                    ENVIA_TRAMA (FRMR)
                    /* envía FRMR y pasa a la fase
                    de desconexión */
                    Desconexión = true
                end_if
            else
                /* trama de información que llega por
                primera vez */
                /*  $N(S) = V(R)$  */

                if N(S) viene en secuencia
                    then
                        Guardar paquete en el buffer
                        en el lugar correspondiente
                        if módulo  $N(N(R) - 1)$  pertenece ventana emisión
                            then
                                /* confirma y modifica ventana */
                                Modificar ventana de emisión a  $N(R) - 1$ 
                                if TimerE fue activado con
                                la última trama confirmada
                                    then
                                        /* desactivar el timer de emisión
                                        porque se confirmaron todas
                                        las tramas */
                                        TimerE := off

                                    end_if
                                end_if
                            end_if

```



```

end_if
if timerR = off
then
    timer := on
end_if
else
    If N(S) cae en la ventana
    then
        Guardar paquete en el buffer
        en el lugar correspondiente
        If módulo N(N(R) - 1) pertenece ventana emisión
        then
            Modificar ventana de emisión a N(R) - 1
            if TimerE fue activado con
            la última trama confirmada
            then
                TimerE := off
            end_if
        end_if
        if trama módulo N(N(S) - 1) no llegó
        then
            ENVIA_TRAMA (O)
        end_if
    else
        ENVIA_TRAMA (FRMR)
        Desconexión = true
    end_if
end_if
otherwise
    ENVIA_TRAMA (FRMR)
    Desconexión = true
end_case
else
    mensajeR = EMPTY
end /* procedure RECIBE_TRAMA */

```

/* el timer recepción está desactivado */
/* lo activa */
/* no viene en secuencia */
/* viene fuera de secuencia pero cae en la ventana */
/* confirma y modifica ventana */
/* desactivar el timer de emisión porque se confirmaron todas las tramas */
/* no llegó la trama anterior a N(S) */
/* envía O para las tramas faltantes */
/* viene fuera de secuencia y N(S) no cae en la ventana */
/* envía FRMR y pasa a la fase de desconexión */
/* trama no standard */
/* envía FRMR y pasa a la fase de desconexión */
/* error de CRC en trama */

Procedure CONEXION

```

begin
    TimerE = off
    i = 1
    repeat
        if (not EMPTY ch)
        then
            RECEIVE ch (mensajeR)
            case mensajeR of
                SABM : If (está en condiciones de conectarse)
                then
                    ENVIA_TRAMA (UA)
                    conectar = true
            end_case
        end_if
    end_repeat

```

/* Escucha el canal y verifica si existe pedido de conexión */
/* recibe SABM */
/* se conecta */
/* envía UA */

```

        else
            ENVIA_TRAMA (DM)
        end_if
        UA : if mensajeS = SABM
            then
                conexión_establecida = true
            end_if
        end_case
    end_if

    if (quiere conectarse) and (not conexión_establecida)
        and (not conectar)
    then
        if (i<=n)
            then
                if timerE = off
                    then
                        timerE = on
                        ENVIA_TRAMA (SABM)
                        i = i+1
                    end_if
                else
                    Resuelve capas superiores
                end_if
            end_if
        until (i>n) or (conexión_establecida) or (conectar)

        if conectar
            then
                if (not EMPTY ch)
                    then
                        RECIBE_TRAMA
                        if (mensajeR = SABM)
                            then
                                ENVIA_TRAMA (UA)
                            else
                                if (mensajeR <> EMPTY)
                                    and (mensajeR <> DISC)
                                then
                                    conexión_establecida = true
                                end_if
                            end_if
                        end_if
                    end_if
                end_if
            end_if

        end /* Procedure CONEXION */

```

Procedure TRANSFERENCIA

```

evento = (Recibiendo, Enviando, Ultimo, Timeout_R,
          CapaRedOciosa, Timeout_E, Capa_Enlace_Ocupada,
          Atención_Concedida, Enviado)

```

```

Procedure RECIBE
begin
  if (not EMPTY ch)
    then                                     /* escucha y recibe algo */
      RECIBE_TRAMA
    end_if
  end /* Procedure RECIBE */

Procedure ENVIA
begin
  if (Capa de Red está habilitada)
    and (not Recibí_RNR)
    and (hay lugar en la ventana)
    then                                     /* se puede enviar trama/s de info */
      Obtiene nuevo paquete
      ENVIA_TRAMA (info)                     /* envía trama de información */
    end_if
  end /* Procedure ENVIA */

Procedure ULTIM
begin
  Desconexión = true                         /* no hay más que hacer */
  ENVIA_TRAMA (DISC)                         /* pasa a la fase de desconexión */
end /* Procedure ULTIM */

Procedure TIMEOUT_RECEPTOR
begin
  if TimerR = off
    then                                     /* venció el timer de recepción */
      if (not Recibí_RNR)
        then                               /* está habilitado el envío */
          if (hay tramas pendientes de confirmación)
            then                           /* queda/n trama/s pendiente/s de confirmación */
              ENVIA_TRAMA (RR)             /* envía RR */
            end_if
          end_if
        end_if
      end_if
  end /* Procedure TIMEOUT_RECEPTOR */

Procedure CAPA_RED_OCIOSA
begin
  if (Capa Red está ociosa) and (not Recibí_RNR)
    then                                     /* el envío está habilitado y la Capa de Enlace está ociosa */
      ENVIA_TRAMA (RR)                     /* envía RR */
    end_if
  end /* Procedure CAPA_RED_OCIOSA */

Procedure TIMEOUT_EMITOR
begin
  if (Buffer_Emisión <> EMPTY)
    then                                     /* hay trama/s pendiente/s de confirmación */

```

```

if TimerE = off
then
    if (not Recibí_RNR)
    then
        if Contador de Retransm. <= n
            retransmisiones
        then
            Contador de Retransm. :=
            Contador de Retrans. + 1
            ENVIA_TRAMA (REQ)
            VREQ := false
            Envie_REQ := true
            TimerE := on
        else
            ENVIA_TRAMA (DISC)
            Desconexion = true
        end_if
    end_if
end_if
end /* procedure TIMEOUT_EMITOR */

```

/* venció el timer de emisión */
 /* el envío está habilitado */
 /* testeo de la cantidad de retrans-
 misiones del REQ */
 /* envía REQ */
 /* activa nuevamente el timer de
 emisión */
 /* hubo más de k retransmisiones para
 el REQ actual */
 /* envía DISC y pasa a la fase de
 desconexión */

wait (evento)

repeat

case evento **of**

```

    Recibiendo      : Recibe
    Enviando        : Envía
    Ultimo          : Ultim
    Timeout_R       : TimeOut_Receptor
    CapaRedOciosa   : Capa_Red_Ociosa
    Timeout_E       : TimeOut_Emisor

```

end_Case

if evento = Capa_Enlace_Ocupada

then

```

    Envie_RNR = true
    ENVIA_TRAMA(RNR)
    Deshabilita_Capa_Red

```

end_if

until (ENVIA_RNR) or Desconexión

if not Desconexión

then

if evento = Atención_Concedida

then

```

    Habilita_Capa_Red
    if (hay trama pendiente de retransmisión)
    then

```

```

        ENVIA_TRAMA (O)

```

else

```

        ENVIA_TRAMA (RR)

```

end_if

Envie_RNR = false

/* la Capa de Enlace está ocupada */

/* envía RNR */

/* se envió RNR anteriormente */

/* se desocupa al Capa de Enlace */

/* hay trama pendiente de retransmisión
 */

/* envía O */

/* no hay trama pendiente de
 aceptación */

/* envía RR */

```

else
    If not EMPTY ch
        then
            RECEIVE ch (mensaje)
            ENVIA_TRAMA (RNR)
            mensajeR = EMPTY
        end_if
    end_if
end_if
end /* Procedure TRANSFERENCIA */

Procedure DESCONEXION
If not EMPTY ch
then
    RECEIVE (mensajeR)
    case mensajeS of
        FRMR : if (mensajeR = SABM) or (mensajeR = DISC)
            then
                Final := true
            else
                If (mensajeR <> DM)
                    then
                        ENVIA_TRAMA (FRMR)
                    end_if
                DISC : if (mensajeR = SABM) or (mensajeR = UA)
                    then
                        Final := true
                        if (mensajeR = SABM)
                            then
                                If está en condiciones de conectarse
                                    then
                                        ENVIA_TRAMA (UA)
                                        Conectar = true
                                    else
                                        ENVIA_TRAMA (DM)
                                    end_if
                                end_if
                            else
                                If mensajeR <> DM
                                    then
                                        ENVIA_TRAMA (DISC)
                                    end_if
                                end_case
                            end
                        else
                            if i <= n
                                then
                                    If Timer = off
                                        then
                                            Timer = on
                                            ENVIA_TRAMA (mensajeS)
                                            i := i + 1
                                        end_if
                                    end_if
                                end_if
                            end
                        end
                    end
                end
            end
        end
    end
end

```

/* realiza evento = Enviado, la Capa de Enlace sigue ocupada */

/* recibe algo */

/* envía nuevamente otro RNR */

/* se envió FRMR */

/* se recibe SABM o DISC */

/* sale de la fase de desconexión */

/* se recibe algo que no es ni SABM, ni DISC ni DM */

/* envía nuevamente FRMR */

/* se envió DISC */

/* se recibe SABM o UA */

/* sale de la fase de desconexión */

/* pasa a la fase de conexión */

/* se conecta */

/* envía UA */

/* no se conecta */

/* envía DM */

/* se recibe algo <> a SABM */

/* se recibe algo que no es ni SABM ni DM */

/* envía DISC */

/* no recibió nada */

/* testea la cantidad de retransmisiones de lo enviado */

/* venció el timer */

/* activa el timer */

/* retransmite la trama anterior */

```
        end_if
    else
        Final = true
    end_if
end /* Procedure DESCONEXION */

begin
final := false
while (not conexión_establecida) do
    CONEXIÓN
end_while

conexión_establecida := false
conectar := false
while (not desconexión) do
    TRANSFERENCIA
end_while

desconexión := false
while (not final) do
    DESCONEXIÓN
end_while

end

end_do

end.
```

/* las retransmisiones sobrepasaron las n */
/* sale de la fase de desconexión */

/* fase del establecimiento de la conexión */

/* fase de transferencia de información */

/* fase de desconexión */

Apéndice C

TESIS.SIM

C. Tesis.Sim

Esta es la presentación vectorial del programa TESIS.SIM que se realizó bajo SIMNET v.135. Se adjunta al final el gráfico de la RdP asociada. Como en el caso de las ejecuciones, se obvió cualquier tipo de modificación.

La parte vectorial de TESIS.SIM consta de dos partes, dedicados a los lugares y luego a las transiciones, indicados en la primera columna como p y t, respectivamente.

Con respecto a los lugares, en la misma aparecen intercalados letras m que señalan a los tokens con los que están inicializados los lugares previos. A continuación se enuncian:

- nombre del lugar
- capacidad
- disciplina de espera :
 - r : aleatorio
 - f : fifo
 - l : lifo
- ubicación del lugar en el diagrama.

Entre las transiciones se observan letras v y z, éstas especifican a los arcos de entrada y de salida de la transición que le preceden. Las otras columnas indican:

- nombre de la transición
- distribución del tiempo de servicio
- prioridad : cuando transiciones concurrentes obtienen la concesión, aquella que tenga la mayor prioridad será la que se va a disparar
- probabilidad: cuando transiciones concurrentes con igual prioridad toman concesión, será seleccionada en base a su probabilidad
- ubicación de la transición en el diagrama.

Los arcos previos a una transición o v tienen las siguientes características:

- nombre del lugar de donde parte el arco
- peso
- tipo de arco.

Los posteriores o z, tienen las columnas:

- nombre del lugar a donde llega el arco
- peso
- tipo de arco.

pp165	1	r	x42 y20
pp162	1	r	x37 y20
pp163	1	r	x42 y24
pp164	1	r	x46 y24
pp158	1	r	x24 y7
pp161	1	r	x22 y26
pp152	1	r	x35 y49
pp153	1	r	x35 y57
pp154	1	r	x35 y65
pp146	1	r	x25 y4
pp147	1	r	x21 y4
pp148	1	r	x25 y9
pp149	1	r	x21 y9
pp150	1	r	x25 y14
pp140	1	r	x50 y50
pp141	1	r	x56 y56
pp142	1	r	x64 y50
pp143	1	r	x70 y56
pp144	1	r	x78 y50
pp145	1	r	x84 y56
pp137	1	r	x28 y65
pp138	1	r	x28 y57
pp139	1	r	x28 y49
pp133	2	r	x16 y18
pp134	1	r	x16 y16
pp135	1	r	x20 y16
pp136	1	r	x14 y19

pp13	1	r	x49 y2
pp5	1	r	x3 y5
m1	1		
pp9	1	r	x2 y49
pp130	1	r	x24 y44
pp131	1	r	x23 y45
pp132	1	r	x24 y46
pp129	1	r	x12 y27
pp127	3	r	x28 y26
pp128	1	r	x39 y23
pp121	1	r	x30 y48
pp122	1	r	x37 y52
pp123	1	r	x30 y56
pp124	1	r	x37 y60
pp125	1	r	x30 y64
pp126	1	r	x37 y68
pp117	1	r	x8 y32
pp118	1	r	x8 y47
pp119	1	r	x5 y44
pp120	1	r	x13 y33
pp116	1	r	x14 y4
pp115	1	r	x9 y4
pp8	1	r	x4 y29
pp114	2	r	x27 y43
pp7	1	r	x3 y27
m1	1		

p p113	1	r	x41 y14
p p111	1	r	x41 y4
p p112	1	r	x41 y9
p p103	2	r	x42 y7
p p104	2	r	x42 y12
p p105	2	r	x42 y17
p p106	1	r	x48 y5
p p107	1	r	x62 y5
p p108	1	r	x76 y5
p p14	1	r	x63 y2
p p3	1	r	x2 y30
p p91	1	r	x52 y68
p p92	1	r	x54 y67
p p93	1	r	x58 y68
p p94	1	r	x60 y67
p p95	1	r	x66 y68
p p96	1	r	x68 y67
p p97	1	r	x72 y68
p p98	1	r	x74 y67
p p99	1	r	x80 y68
p p100	1	r	x82 y67
m 1	1		
p p101	1	r	x86 y68
p p102	1	r	x88 y67
p p80	1	r	x82 y53
p p81	1	r	x68 y53
p p82	1	r	x54 y53
p p83	1	r	x14 y25
p p84	1	r	x13 y30
m 1	1		
p p85	1	r	x16 y22
p p86	2	r	x16 y24
p p87	1	r	x16 y28
p p88	2	r	x16 y30
m 1	1		
p p89	1	r	x20 y22
p p90	1	r	x20 y28
p req	1	r	x25 y26
p p10	1	r	x3 y51
p p79	2	r	x9 y26
p p75	1	r	x8 y7
p p76	1	r	x8 y29
p p77	1	r	x7 y51
p p2	1	r	x2 y8
p p63	1	r	x53 y47
p p64	1	r	x59 y53
p p17	1	r	x53 y19
p p18	1	r	x67 y19
p p67	1	r	x67 y47
p p68	1	r	x73 y53
p p19	1	r	x81 y19
m 1	1		
p p70	1	r	x81 y47
p p15	1	r	x77 y2
p p72	1	r	x87 y53

p p27	1	r	x63 y36
p p25	1	r	x65 y34
p p6	1	r	x4 y7
p p58	1	r	x51 y7
p p59	1	r	x65 y7
p p60	1	r	x79 y7
p p1	1	r	x1 y4
m 1	1		
p p46	1	r	x49 y15
p p47	1	r	x63 y15
p p48	1	r	x77 y15
p p28	1	r	x77 y36
p p50	1	r	x50 y11
p p51	1	r	x64 y11
p p52	1	r	x78 y11
p p21	1	r	x65 y24
p p54	1	r	x54 y15
p p55	1	r	x69 y15
p p56	1	r	x82 y15
p p26	1	r	x49 y36

* envio de paquetes

t t1	n(2,0.001)		x2 y6
v p5	k(1)	1	
v p1	k(1)	1	
z p6	k(1)	1	
z p2	k(1)	1	
t t2	n(2,0.001)		x2 y28
v p7	k(1)	1	
v p2	k(1)	1	
z p8	k(1)	1	
z p3	k(1)	1	
t t3	n(2,0.001)		x1 y50
v p9	k(1)	1	
v p3	k(1)	1	
z p10	k(1)	1	
z p1	k(1)	1	

* envio de tramas

t t9	n(3,0.001)	20	1	x6 y7
v p6	k(1)	1		
z p115	k(1)	1		
z p75	k(1)	1		
z p79	k(1)	1		
t t10	n(3,0.001)	20	1	x6 y29
v p8	k(1)	1		
z p117	k(1)	1		
z p76	k(1)	1		
z p79	k(1)	1		

t t11	n(3,0.001)	20	1	x5 y51
v p10	k(1)	1		
z p118	k(1)	1		
z p77	k(1)	1		
z p79	k(1)	1		

* procesamiento de tramas

t t13	k(0)			x49 y21
v p46	k(1)	1		
v p19	k(1)	1		
z p26	k(1)	1		
z p21	k(1)	1		
z p17	k(1)	1		
z p54	k(1)	1		

t t14	k(0)			x63 y21
v p17	k(1)	1		
v p47	k(1)	1		
z p27	k(1)	1		
z p18	k(1)	1		
z p55	k(1)	1		
z p21	k(1)	1		

t t15	k(0)			x77 y21
v p18	k(1)	1		
v p48	k(1)	1		
z p28	k(1)	1		
z p19	k(1)	1		
z p56	k(1)	1		
z p21	k(1)	1		

* confirmaciones

t t25	n(1,0.002)	10	1	x52 y45
v p25	k(1)	1		
v p26	k(1)	1		
v p27	k(0)	1	<	
z p63	k(1)	1		
z p91	k(1)	1		

t t26	n(1,0.002)	0	1	x58 y45
v p25	k(1)	1		
v p27	k(1)	1		
v p26	k(1)	1		
z p64	k(1)	1		
z p93	k(1)	1		

t t28	n(1,0.002)	10	1	x66 y45
v p27	k(1)	1		
v p25	k(1)	1		

v p28	k(0)	1	<	
z p67	k(1)	1		
z p95	k(1)	1		
t t30	n(1,0.002)	0	1	x72 y45
v p28	k(1)	1		
v p27	k(1)	1		
v p25	k(1)	1		
z p68	k(1)	1		
z p97	k(1)	1		
t t32	n(1,0.002)	10	1	x80 y45
v p28	k(1)	1		
v p25	k(1)	1		
v p26	k(0)	0	<	
z p70	k(1)	1		
z p99	k(1)	1		
t t34	n(1,0.002)	0	1	x86 y45
v p26	k(1)	1		
v p25	k(1)	1		
v p28	k(1)	1		
z p72	k(1)	1		
z p101	k(1)	1		

* pedidos de retransmision

t t37	n(1,0.001)	10	1	x44 y4
v p17	k(0)	1	<	
v p50	k(0)	1	<	
v p59	k(1)	1		
z p103	k(1)	1		
z p111	k(1)	1		
t t39	n(1,0.001)	10	1	x44 y9
v p51	k(0)	1	<	
v p18	k(0)	1	<	
v p60	k(1)	1		
z p104	k(1)	1		
z p112	k(1)	1		
t t41	n(1,0.001)	10	1	x44 y1
v p52	k(0)	1	<	
v p19	k(0)	1	<	
v p58	k(1)	1		
z p113	k(1)	1		
z p105	k(1)	1		
t t67	k(0)			x80 y11
v p52	k(1)	1		
v p56	k(1)	1		

* llegada de tramas de info al receptor

t t61	n(1.5,0.001)			x49 y6
-------	--------------	--	--	--------

v p13	k(1)	1		
z p106	k(1)	1		
z p46	k(1)	1		
z p50	k(1)	1		
z p58	k(1)	1		
t t62	n(1.5,0.001)			x63 y6
v p14	k(1)	1		
z p107	k(1)	1		
z p47	k(1)	1		
z p51	k(1)	1		
z p59	k(1)	1		
t t63	n(1.5,0.001)			x77 y6
v p15	k(1)	1		
z p108	k(1)	1		
z p48	k(1)	1		
z p52	k(1)	1		
z p60	k(1)	1		
t t65	k(0)			x52 y11
v p50	k(1)	1		
v p54	k(1)	1		
t t66	k(0)			x66 y11
v p51	k(1)	1		
v p55	k(1)	1		
t t69	k(0)	0	1	x53 y7
v p58	k(1)	1		
v p46	k(0)	1	<	
v p19	k(0)	1	<	
t t70	k(0)	0	1	x67 y7
v p59	k(1)	1		
v p47	k(0)	1	<	
v p17	k(0)	1	<	
t t71	k(0)	0	1	x81 y7
v p60	k(1)	1		
v p48	k(0)	1	<	
v p18	k(0)	1	<	
* timer del receptor				
t t73	k(6)	0	1	x65 y31
v p21	k(1)	1		
v p25	k(0)	1	<	
z p25	k(1)	1		
* confirm. en linea				
t t76	n(2,0.001)	1	0.898	x87 y56
v p72	k(1)	1		

z p164	k(1)	1		
z p145	k(1)	1		
t t78	n(2,0.001)	1	0.898	x81 y50
v p70	k(1)	1		
z p164	k(1)	1		
z p144	k(1)	1		
t t80	n(2,0.001)	1	0.898	x73 y56
v p68	k(1)	1		
z p164	k(1)	1		
z p143	k(1)	1		
t t81	n(2,0.001)	1	0.898	x67 y50
v p67	k(1)	1		
z p164	k(1)	1		
z p142	k(1)	1		
t t84	n(2,0.001)	1	0.898	x59 y56
v p64	k(1)	1		
z p164	k(1)	1		
z p141	k(1)	1		
t t85	n(2,0.001)	1	0.898	x53 y50
v p63	k(1)	1		
z p164	k(1)	1		
z p140	k(1)	1		

*** perdidas de tramas**

t t87	k(0)	1	0.1	x8 y8
v p75	k(1)	1		
t t89	k(0)	1	0.1	x8 y30
v p76	k(1)	1		
t t91	k(0)	1	0.1	x7 y52
v p77	k(1)	1		

*** tramas en linea**

t t88	n(2,0.001)	1	0.898	x10 y29
v p76	k(1)	1		
z p14	k(1)	1		
t t90	n(2,0.001)	1	0.898	x9 y51
v p77	k(1)	1		
z p15	k(1)	1		
t t86	n(2,0.001)	1	0.898	x10 y7
v p75	k(1)	1		
z p13	k(1)	1		

*** trama que no pone en marcha el timer**

t t17	k(0)	10	1	x64 y25
v p21	k(1)	1		
v p17	k(1)	1	>	
t t93	k(0)	10	1	x66 y26
v p21	k(1)	1		
v p18	k(1)	1	<	
t t92	k(0)	10	1	x67 y25
v p21	k(1)	1		
v p19	k(1)	1	>	

* perdida de trama

t t94	k(0)	1	0.1	x82 y52
v p70	k(1)	1		
z p80	k(1)	1		
t t95	k(0)	0	1	x83 y53
v p63	k(1)	1		
v p80	k(1)	1		
z p72	k(1)	1		
t t96	k(0)	1	0.1	x68 y52
v p67	k(1)	1		
z p81	k(1)	1		
t t97	k(0)	0	1	x69 y53
v p70	k(1)	1		
v p81	k(1)	1		
z p68	k(1)	1		
t t98	k(0)	1	0.1	x54 y52
v p63	k(1)	1		
z p82	k(1)	1		
t t99	k(0)	0	1	x55 y53
v p67	k(1)	1		
v p82	k(1)	1		
z p64	k(1)	1		
t t100	k(0)	10	1	x55 y54
v p82	k(1)	1		
t t101	k(0)	1	0.1	x60 y53
v p64	k(1)	1		
t t102	k(0)	10	1	x69 y54
v p81	k(1)	1		
t t103	k(0)	1	0.1	x74 y53
v p68	k(1)	1		
t t104	k(0)	10	1	x83 y54
v p80	k(1)	1		



BIBLIOTECA
MATEMÁTICA
U. N. C.

t t105	k(0)	1	0.1	x88 y53
v p72	k(1)	1		

* administradores de timers

t t106	k(0)			x14 y23
v p129	k(1)	1		
v p136	k(1)	1		
z p86	k(1)	1		
z p85	k(1)	1		
z p83	k(1)	1		

t t107	k(0)			x14 y29
v p129	k(1)	1		
v p83	k(1)	1		
z p88	k(1)	1		
z p87	k(1)	1		
z p84	k(1)	1		

t t187	k(0)			x14 y17
v p129	k(1)	1		
v p84	k(1)	1		
z p134	k(1)	1		
z p133	k(1)	1		
z p136	k(1)	1		

* timer de emision

t tim1	k(17)			x18 y22
v p85	k(1)	1		
v p133	k(1)	1		
z p89	k(1)	1		

t tim2	k(17)			x18 y28
v p86	k(1)	1		
v p87	k(1)	1		
z p90	k(1)	1		

t tim3	k(17)			x18 y16
v p88	k(1)	1		
v p134	k(1)	1		
z p135	k(1)	1		

* timer off

t t110	k(0)	10	1	x20 y21
v p127	k(1)	1		
v p89	k(1)	1		

t t111	k(0)	10	1	x20 y27
v p90	k(1)	1		
v p127	k(1)	1		

t t190	k(0)	10	1	x20 y15
v p135	k(1)	1		
v p127	k(1)	1		

t time1	k(0)	0	1	x21 y23
v p86	k(1)	1		
v p89	k(1)	1		
z p161	k(1)	1		
z p86	k(1)	1		

t time2	k(0)	0	1	x21 y29
v p88	k(1)	1		
v p90	k(1)	1		
z p161	k(1)	1		
z p88	k(1)	1		

t time3	k(0)	0	1	x21 y17
v p135	k(1)	1		
v p133	k(1)	1		
z p133	k(1)	1		
z p161	k(1)	1		

t t115	k(0)			x23 y26
v p127	k(1)	1		
v p161	k(1)	1		
z req	k(1)	1		

* confirmac. frente a un REQ

t t116	k(0)	10	1	x47 y48
v p121	k(1)	1		
v p92	k(1)	1	>	
z p63	k(1)	1		

t t117	k(0)	10	1	x49 y52
v p122	k(1)	1		
v p94	k(1)	1	>	
z p64	k(1)	1		

t t118	k(0)	10	1	x47 y56
v p123	k(1)	1		
v p96	k(1)	1	>	
z p67	k(1)	1		

t t119	k(0)	10	1	x49 y60
v p124	k(1)	1		
v p98	k(1)	1	>	
z p68	k(1)	1		

t t120	k(0)	10	1	x47 y64
v p125	k(1)	1		
v p100	k(1)	1	>	
z p70	k(1)	1		

t t121	k(0)	10	1	x49 y68
--------	------	----	---	---------

v p126	k(1)	1	
v p102	k(1)	1	>
z p72	k(1)	1	

* indicadores de ultima confirmacion en receptor

t t122	k(0)		x51 y70
v p98	k(1)	1	
v p91	k(1)	1	
z p92	k(1)	1	

t t123	k(0)		x53 y70
v p100	k(1)	1	
v p91	k(1)	1	
z p92	k(1)	1	

t t124	k(0)		x57 y70
v p98	k(1)	1	
v p93	k(1)	1	
z p94	k(1)	1	

t t125	k(0)		x59 y70
v p100	k(1)	1	
v p93	k(1)	1	
z p94	k(1)	1	

t t126	k(0)		x65 y70
v p92	k(1)	1	
v p95	k(1)	1	
z p96	k(1)	1	

t t127	k(0)		x67 y70
v p102	k(1)	1	
v p95	k(1)	1	
z p96	k(1)	1	

t t128	k(0)		x71 y70
v p92	k(1)	1	
v p97	k(1)	1	
z p98	k(1)	1	

t t129	k(0)		x73 y70
v p102	k(1)	1	
v p97	k(1)	1	
z p98	k(1)	1	

t t130	k(0)		x79 y70
v p94	k(1)	1	
v p99	k(1)	1	
z p100	k(1)	1	

t t131	k(0)		x81 y70
v p96	k(1)	1	
v p99	k(1)	1	
z p100	k(1)	1	

t t132	k(0)			x85 y70
v p94	k(1)	1		
v p101	k(1)	1		
z p102	k(1)	1		

t t133	k(0)			x87 y70
v p96	k(1)	1		
v p101	k(1)	1		
z p102	k(1)	1		

*** llegada de la trama retransmitida al receptor**

t t137	k(0)	0	1	x44 y7
v p103	k(1)	1		
v p106	k(1)	1		

t t139	k(0)	0	1	x44 y12
v p107	k(1)	1		
v p104	k(1)	1		

t t141	k(0)	0	1	x44 y17
v p108	k(1)	1		
v p105	k(1)	1		

t t138	k(0)	10	1	x47 y5
v p106	k(1)	1		

t t140	k(0)	10	1	x61 y5
v p107	k(1)	1		

t t142	k(0)	10	1	x75 y5
v p108	k(1)	1		

*** perdida y puesta en linea de pedido de retransmision**

t t146	k(0)	1	0.1	x41 y5
v p111	k(1)	1		

t t147	n(2,0.002)	1	0.898	x28 y4
v p111	k(1)	1		
z p146	k(1)	1		

t t148	n(2,0.002)	1	0.898	x34 y9
v p112	k(1)	1		
z p148	k(1)	1		

t t149	k(0)	1	0.1	x41 y10
v p112	k(1)	1		

t t150	k(0)	1	0.1	x41 y15
v p113	k(1)	1		

t t151	n(2,0.002)	1	0.898	x28 y14
--------	------------	---	-------	---------

v p113	k(1)	1		
z p150	k(1)	1		
t t152	k(0)			x11 y4
v p116	k(1)	1		
v p115	k(1)	1		
t t153	k(0)			x7 y4
v p115	k(1)	1		
v p114	k(1)	1		
* REQ				
t t154	n(2,0.001)	5	1	x12 y9
v p127	k(0)	1	<	
v p115	k(1)	1	>	
v req	k(1)	1		
z p162	k(1)	1		
z p128	k(1)	1		
z p121	k(1)	1		
t t155	n(2,0.001)	5	1	x8 y34
v p127	k(0)	1	<	
v p117	k(1)	1	>	
v req	k(1)	1		
z p162	k(1)	1		
z p128	k(1)	1		
z p123	k(1)	1		
t t156	k(0)			x11 y35
v p117	k(1)	1		
v p114	k(1)	1		
t t157	k(0)			x11 y33
v p117	k(1)	1		
v p120	k(1)	1		
t t158	n(2,0.001)	0	1	x8 y21
v p127	k(0)	1	<	
v p115	k(1)	1	>	
v p117	k(1)	1	>	
v req	k(1)	1		
z p162	k(1)	1		
z p128	k(1)	1		
z p122	k(1)	1		
t t159	k(0)			x5 y45
v p119	k(1)	1		
v p118	k(1)	1		
t t160	n(2,0.001)	0	1	x11 y47
v p127	k(0)	1	<	
v p118	k(1)	1	>	
v req	k(1)	1		

v p117	k(1)	1	>	
z p162	k(1)	1		
z p128	k(1)	1		
z p124	k(1)	1		
t t161	k(0)			x8 y46
v p118	k(1)	1		
v p114	k(1)	1		
t t163	n(2,0.001)	5	1	x10 y43
v p127	k(0)	1	<	
v p118	k(1)	1	>	
v req	k(1)	1		
z p162	k(1)	1		
z p128	k(1)	1		
z p125	k(1)	1		
t t164	n(2,0.001)	0	1	x17 y37
v p127	k(0)	1	<	
v p115	k(1)	1	>	
v p118	k(1)	1	>	
v req	k(1)	1		
z p162	k(1)	1		
z p128	k(1)	1		
z p126	k(1)	1		

*** perdida de REQ**

t t165	k(0)	20	1	x30 y49
v p121	k(1)	1		
t t166	k(0)	20	1	x37 y53
v p122	k(1)	1		
t t167	k(0)	20	1	x30 y57
v p123	k(1)	1		
t t168	k(0)	20	1	x37 y61
v p124	k(1)	1		
t t169	k(0)	20	1	x30 y65
v p125	k(1)	1		
t t170	k(0)	20	1	x37 y69
v p126	k(1)	1		

*** REQ enviados ante tramas pendientes**

t t171	k(0)	10	1	x30 y47
v p92	k(0)	1	<	
v p121	k(1)	1		
z p163	k(1)	1		
z p139	k(1)	1		
t t172	k(0)	10	1	x37 y51

v p92	k(0)	1	<	
v p94	k(0)	1	<	
v p122	k(1)	1		
z p163	k(1)	1		
z p152	k(1)	1		
z p116	k(1)	1		
z p6	k(1)	1		

t t173	k(0)	10	1	x30 y55
v p96	k(0)	1	<	
v p123	k(1)	1		
z p163	k(1)	1		
z p138	k(1)	1		

t t174	k(0)	10	1	x37 y59
v p96	k(0)	1	<	
v p98	k(0)	1	<	
v p124	k(1)	1		
z p163	k(1)	1		
z p153	k(1)	1		
z p120	k(1)	1		
z p8	k(1)	1		

t t175	k(0)	10	1	x30 y63
v p100	k(0)	1	<	
v p125	k(1)	1		
z p163	k(1)	1		
z p137	k(1)	1		

t t176	k(0)	10	1	x37 y67
v p100	k(0)	1	<	
v p102	k(0)	1	<	
v p126	k(1)	1		
z p163	k(1)	1		
z p154	k(1)	1		
z p119	k(1)	1		
z p10	k(1)	1		

* puesta del timer por REQ cuando no hay tramas de info

t t177	k(0)	0	1	x26 y24
v p128	k(1)	1		
z p79	k(1)	1		

t t180	k(0)			x10 y26
v p79	k(1)	1		
z p129	k(1)	1		
z p127	k(1)	1		

t t181	k(0)			x9 y48
v p131	k(1)	1		
v p118	k(1)	1		

t t182	k(0)			x6 y34
v p130	k(1)	1		
v p117	k(1)	1		

t t183	k(0)			x11 y5
v p132	k(1)	1		
v p115	k(1)	1		

* pedido de retransmision que se reemite frente a un REQ

t t184	k(0)	0	1	x26 y8
v p103	k(1)	1		
v p122	k(1)	1		
z p111	k(1)	1		
z p103	k(1)	1		

t t185	k(0)	0	1	x26 y13
v p104	k(1)	1		
v p124	k(1)	1		
z p112	k(1)	1		
z p104	k(1)	1		

t t186	k(0)	0	1	x26 y18
v p105	k(1)	1		
v p126	k(1)	1		
z p113	k(1)	1		
z p105	k(1)	1		

* confirmaciones que se tienen en cuenta y otras no ante un REQ y una trama de
* info enviada posteriormente

t t191	k(0)	10	1	x26 y64
v p137	k(1)	1		
z p119	k(1)	1		
z p10	k(1)	1		

t t192	k(0)	0	1	x25 y66
v p115	k(1)	1	>	
v p137	k(1)	1		

t t193	k(0)	10	1	x26 y56
v p138	k(1)	1		
z p120	k(1)	1		
z p8	k(1)	1		

t t194	k(0)	0	1	x25 y58
v p118	k(1)	1	>	
v p138	k(1)	1		

t t195	k(0)	10	1	x26 y48
v p139	k(1)	1		
z p116	k(1)	1		
z p6	k(1)	1		

t t196	k(0)	0	1	x25 y50
--------	------	---	---	---------

v p117 k(1) 1 >
v p139 k(1) 1

* confirmaciones que corren ventana y otras que no

t t197 k(0) 10 1 x50 y49
v p140 k(1) 1

t t198 k(0) 0 1 x47 y50
v p140 k(1) 1
v p115 k(1) 1 >
z p132 k(1) 1
z p9 k(1) 1
z p158 k(1) 1

t t199 k(0) 10 1 x56 y55
v p141 k(1) 1

t t200 k(0) 0 1 x53 y56
v p141 k(1) 1
v p115 k(1) 1 >
v p117 k(1) 1 >
z p9 k(1) 1
z p5 k(1) 1
z p114 k(2) 1
z p158 k(1) 1

t t201 k(0) 10 1 x64 y49
v p142 k(1) 1

t t202 k(0) 0 1 x61 y50
v p142 k(1) 1
v p117 k(1) 1 >
z p130 k(1) 1
z p5 k(1) 1
z p158 k(1) 1

t t203 k(0) 10 1 x70 y55
v p143 k(1) 1

t t204 k(0) x67 y56
v p143 k(1) 1
v p117 k(1) 1 >
v p118 k(1) 1 >
z p114 k(2) 1
z p7 k(1) 1
z p5 k(1) 1
z p158 k(1) 1

t t205 k(0) 10 1 x78 y49
v p144 k(1) 1

t t206 k(0) 0 1 x75 y50
v p144 k(1) 1

v p118	k(1)	1	>	
z p131	k(1)	1		
z p7	k(1)	1		
z p158	k(1)	1		
t t207	k(0)	10	1	x84 y55
v p145	k(1)	1		
t t208	k(0)	0	1	x81 y56
v p145	k(1)	1		
v p115	k(1)	1	>	
v p118	k(1)	1	>	
z p114	k(2)	1		
z p7	k(1)	1		
z p9	k(1)	1		
z p158	k(1)	1		

* llegada de un pedido de retransmision al emisor

t t210	k(0)	10	1	x23 y4
v p146	k(1)	1		
z p149	k(1)	1		
z p116	k(1)	1		
z p6	k(1)	1		
t t211	k(0)	5	1	x23 y6
v p158	k(1)	1		
v p147	k(1)	1		
t t212	k(0)	10	1	x22 y7
v p158	k(1)	1		
t t215	k(0)	10	1	x23 y9
v p148	k(1)	1		
z p120	k(1)	1		
z p8	k(1)	1		
z p149	k(1)	1		
t t220	k(0)	10	1	x23 y14
v p150	k(1)	1		
z p119	k(1)	1		
z p10	k(1)	1		
z p149	k(1)	1		

* retransmisiones de 2 tramas, respetar el orden de salida

t t224	n(2,0.001)			x35 y47
v p152	k(1)	1		
z p120	k(1)	1		
z p8	k(1)	1		
t t225	n(2,0.001)			x35 y55
v p153	k(1)	1		
z p119	k(1)	1		

z p10 k(1) 1

t t226 n(2,0.001) x35 y63

v p154 k(1) 1

z p116 k(1) 1

z p6 k(1) 1

* conf. de una sola trama que deben volver a emitirse ante un REQ

t t227 k(0) x43 y54

v p122 k(1) 1

v p92 k(1) 1 >

z p63 k(1) 1

t t228 k(0) x43 y62

v p124 k(1) 1

v p96 k(1) 1 >

z p67 k(1) 1

t t229 k(0) x43 y70

v p126 k(1) 1

v p100 k(1) 1 >

z p70 k(1) 1

t t230 k(0) 10 1 x25 y27

v req k(1) 1

t t232 k(0) 10 1 x20 y9

v p149 k(1) 1

t t234 k(0) x41 y23

v p165 k(1) 1

v p163 k(1) 1

t t235 k(0) 0 1 x45 y22

v p165 k(1) 1

v p164 k(1) 1

t t236 k(0) 10 1 x47 y23

v p164 k(1) 1

t t237 k(0) 5 1 x40 y18

v p165 k(1) 1

v p149 k(1) 1

z p147 k(1) 1

t t238 k(0) x25 y12

v p150 k(1) 1

v p147 k(1) 1

t t239 k(0) x25 y7

v p148 k(1) 1

v p147 k(1) 1

t t240	k(0)			x25 y2
v p146	k(1)	1		
v p147	k(1)	1		
t t241	k(0)			x39 y20
v p165	k(0)	1	<	
v p162	k(1)	1		
z p165	k(1)	1		
t t242	k(0)			x42 y19
v p162	k(1)	1	>	
v p165	k(1)	1		
t t243	k(0)	0	1	x44 y19
v p165	k(1)	1	>	
v p147	k(1)	1		



REGLAMENTO A
FACULTAD DE INGENIERÍA
UNLP

CONVOCACION

TES

97/18 p. 1

31-8-05

DE-30

1990

